MAY 1976 $1.50

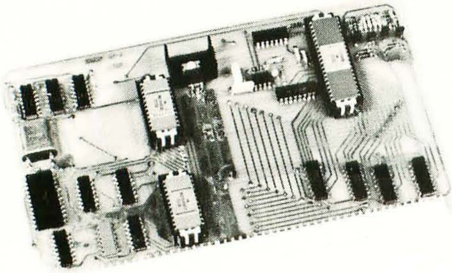
BYTE

the small systems journal

SHOOTING STARS

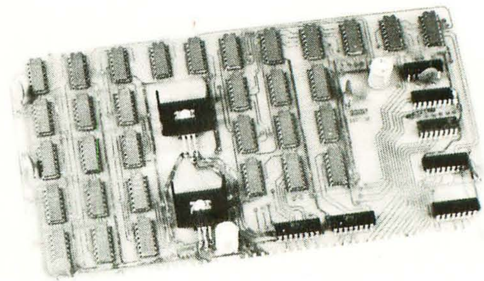


IT ADDS UP–



MP-A processor card. A complete microcomputer system on a single card. It features the "Motorola" MC6800 processor. The powerful memory orientated processor that is rapidly becoming the standard of comparison in the industry. Also on this board is the MCM6830L7 ® Mikbug ROM, MC6810 RAM and the MC14411 baud rate generator. The crystal controlled master clock oscillator and tri-state data bus drivers complete the board. Everything works from a single 5.0 Volt supply.

MP-A. $145.00

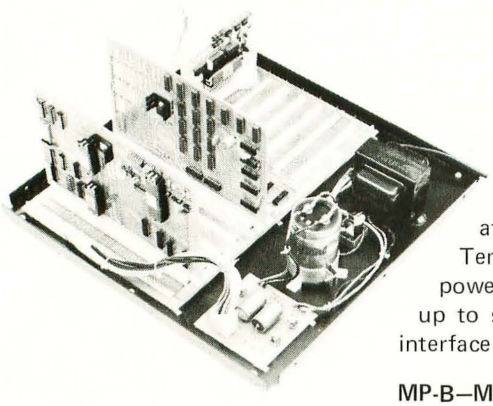


MP-M main memory card. This circuit board contains two blocks of 2,048 words of memory. Each part has its own supply regulator and can operate independently of the other half. The memory chips are the proven reliable 2102 static type. They are capable of cycling fast enough to allow full speed operation of the processor at all times. No waiting for refresh cycles and no problems with "flakey" memories. Only a single 5.0 Volt supply required.

MP-M. $125.00

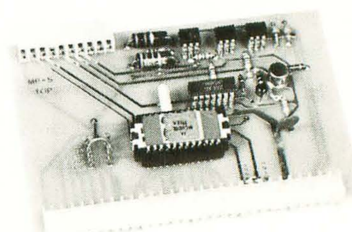
MP-C serial control interface. This programmable interface may be used to connect the computer to either a video terminal or TTY current loop system. Directly compatable with either system. No hardware changes.

MP-C $40.00



Chassis, mother board and power supply. Anodized cabinet with perforated cover for cool operation. Ten amp power supply will power fully expanded system of up to six memory cards and eight interface cards.

MP-B—Mother Board. $40.00
MP-F—Chassis $30.00
MP-P—Power Supply $35.00



TO A GREAT COMPUTER SYSTEM

SwTPC 6800
Computer System

with serial interface and 2,048 words of memory. $395.00



Mikbug® is a registered trademark of Motorola Inc.

☐ **Enclosed is $395 for my SwTPC Computer Kit** ☐ **Send Data**

☐ **or BAC** ______________________ **#** ______________

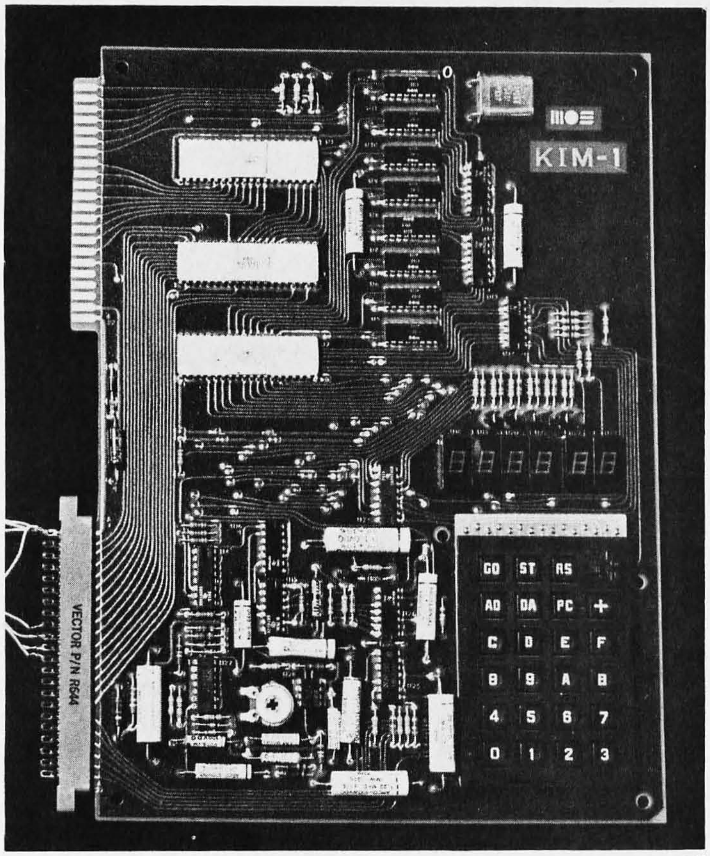
☐ **or MC** ______________________ **Ex Date** ____________

____
NAME

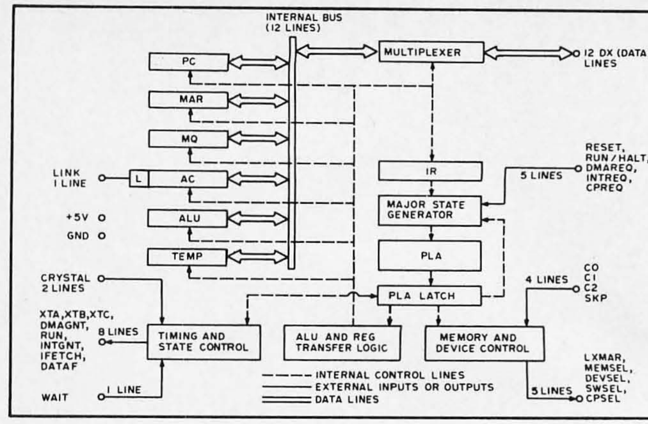
____
ADDRESS

____
CITY STATE ZIP

Southwest Technical Products Corp., Box 32040, San Antonio, Texas 78284



page 8



page 60

In This BYTE

Richard Simpson describes his first **Date With KIM**, the new product from MOS Technology which comes assembled and ready to use. This product, which is the basis of his system, marks the first direct entry of a semiconductor manufacturer into the personal systems field.

Are different microcomputers equivalent? In **n Source**, RD Boudinot presents some excellent background information on multiple sources of components and systems, the mixing of products from different manufacturers and methods of evaluating products for use in a personal computing system.

Of what use is a nice friendly permanent memory? Dale Eichbauer contributes some ideas on the use of **Read Only Memories in Microcomputer Memory Address Space.**

Previous articles have covered programming and uses of some of the simpler fusible link read only memories. But how about erasable ROMs? Roger L Smith provides some **More Information on PROMs** including a method of programming the widely available 1702 parts.

One way to get a hard copy terminal is to use a receive only Teletype unit. Using an inexpensive ASCII keyboard and a UART circuit, Dr George Haller shows how to **Serialize the Bits From Your Mystery Keyboard** and achieve the same function as a keyboard send receive Teletype for about half the cost.

Dissatisfied with toggle switches? Use **An Octal Front Panel** similar to Herman DeMonstoy's design to replace toggle switches with an octal keyboard.

You'll be **SHOOTING STARS** in a fascinating logical game when you implement a version of Willard Nico's program on your computer. On the cover is artist Robert Tinney's impression of a SHOOTING STARS addict.

A simple signal generator might suffice for a radio man, but testing of computers and data communications hardware can require more sophisticated equipment. One such item is a **Serial ASCII Word Generator** such as the design Ronald Finger describes.

How do you take advantage of a decade of software experience? One way is to emulate another computer's architecture as Intersil has done with its IM6100. Robert Nelson describes a **"Chip" Off the Olde PDP-8E** in this first part of a two part article.

Can a computer measure voltages without hundreds of dollars worth of hardware? Of course it can. The secret is to use **Microprocessor Based Analog/Digital Conversion Techniques** of the sort described in Roger Frank's article on a very basic interface.

One of the keys to creating an assembler is defining exactly what the input source language will look like. An appropriate choice which simplifies writing the assembler will greatly speed up the process of implementing the program. In his article on the subject, Gregory Jewell shows how to **Simplify Your Homemade Assembler** using techniques which are applicable to most microcomputers.

In the Queue

BYTE #9

MAY 1976

Foreground

Background

Nucleus

staff

PUBLISHERS
Virginia Peschke
Manfred Peschke
EDITOR
Carl T Helmers, Jr
GENERAL MANAGER
Manfred Peschke
PRODUCTION MANAGER
Judith Havey
PRODUCTION ASSISTANT
Elizabeth Alpaugh
CIRCULATION
Deborah R Luhrs
PUBLISHERS ASSISTANTS
Cheryl Hurd
Carol Nyland
Deena Zealy
ADVERTISING
Elizabeth Alpaugh
Virginia Peschke
TYPOGRAPHY
Custom Marketing Resources, Inc
Goodway Graphics
Mary Lavoie
Taimi Woodward
PHOTOGRAPHY
Ed Crabtree
Custom Marketing Resources, Inc
ART
Bill Morello
PRINTING
Custom Marketing Resources, Inc
The George Banta Company
ASSOCIATES
Dan Fylstra
Don Lancaster
Harold A Mauch
Chris Ryland

BYTE magazine is published monthly by BYTE Publications, Inc., 70 Main St, Peterborough, New Hampshire 03458. Subscription rates are $12 for one year worldwide. Two years, $22. Three years, $30. Second class postage paid at Peterborough New Hampshire 03458 and at additional mailing offices. Phone 603-924-7217. Address editorial correspondence to Editor, BYTE, 70 Main St, Peterborough NH 03458.

Trends in Applications

Editorial by Carl Helmers

Prognostication is like an operational amplifier system . . .

A computer system can be used to perform an old task using a new approach which requires less personal effort or work.

[This editorial is taken from the text of a talk presented at the recent "World Altair Convention" held by MITS Inc in Albuquerque NM.]

Where is the small computer field headed? In order to talk about the future of small computers and their applications, I have to take on the role of a prognosticator, a predictor of future trends and events. Prognostication is an art to which mystical or magical qualities are often attributed, but which in reality is nothing more than a combination of reasoning and imagination based upon observation. The injection of imagination about possible trends and developments makes prognostication a bit different from a narrow linear extrapolation of identified trends. The imagination component is heavily influenced by personal values and philosophies, and represents a feedback of oughts and shoulds into the course of events as they develop. Prognostication is thus a method of extrapolating observed current trends into the future coupled with the prognosticator's opinions of what should be happening. In the terms of the scientist or engineer, prognostication is like an operational amplifier system in which the input signal is the observed set of trends and the feedback network is the prognosticator's personal philosophy and imagination. For example, in predicting the fate of civilization, if one is a congenital pessimist like the members of the Club of Rome, then the predictions will come out claiming disaster and ruin. If one is an optimist about the expanding possibilities created by advancing technology, then a totally different character of prediction will result. When you listen to what I have to say, be warned that I have a definite personal point of view regarding computer technology and its proper uses, and that this shapes the nature of the imagination content and the trends I select to emphasize.

A Point of View

The point of view from which I build my conceptual model of a possible future state of the computer world is the view that the individual person is the most important component part of the human species. It is an observed fact that every great advance made in scientific progress, every great work of art, and every notable human achievement is the result of the work of individual human beings, whether or not the ultimate source of the idea, work or achievement can be identified. What is true for the notable accomplishments is just as true for the ones which may not be individually recorded in history books. It is the individual human being with responsible self interest at heart who discovers new ways to handle old problems, invents new problems and their solutions, creates works of art and leads to an improved way of life. My views of the trends in computing are thus oriented to the ways in which computer systems technology can provide a better and more comfortable existence for you – the individuals who are in the knowledgeable vanguard of the new technology of personal computing. In a sense, one of the most exciting aspects of the present time is the prospect that we – you, I, the rest of civilization – are in the early stages of one of the "golden ages" of the planet's history, a time when art and science are flourishing throughout the more advanced segments of the civilization. Computing is one important characteristic of this current age.

What Are Computers Used For?

So much for the preface. Just what are computer systems used for? How will the characteristics of these uses develop as a result of the constant improvement of hardware and software techniques? At the highest level, I can identify two major facets of the computer system's application:

- A computer system can be used to perform an old task using a new approach which requires less personal effort or work.
- A computer system can be used to accomplish new tasks which were previously unattainable without the "intelligence" of the processor with its stored programs and conditional execution.

In any given application, there is not necessarily a sharp distinction between the two facets of the computer's use. But this view illuminates two interesting aspects of the technology, and can be used in the analysis of a computer's importance to an application. A couple of extreme examples will illustrate what I mean by these facets of computing.

A good example of an old task which can be expedited considerably by use of a computer system is the personal accounting task of balancing a check book. In the modern

American way of life, the check book is one of the most ubiquitous of personal financial instruments. Unless you live dangerously, you balance that check book once a month, whenever the bank statement comes. The method of balancing a check book is a boring procedure which is well defined and nearly universal in its use. By employing a computer system, this boring procedure can be expedited through automation. The method is to use a program with interactive characteristics to enter the data, perform the arithmetic, and – if you have hard copy – give you a record of the transactions on paper. Using such programs, the accuracy of the check book can in general be improved and the time required each month can be considerably reduced. This reduction in time wasted on check books can be put to use in other more enjoyable tasks, thus improving your state of well being and happiness. The essence of this type of a computer application is use of the system as a "busywork eliminator," a term I first heard applied by a long time friend, Ken Hardwick. The busywork is more efficiently performed through automation, thus minimizing the human demands of the work.

A computer system can be used to accomplish new tasks which were previously unattainable without the "intelligence" of the processor with its stored programs and conditional execution.

An example of the new task which could not previously have been accomplished is provided by every highly automated interactive game which is developed and run on a computer system. There is no way that you or I could have played Space War or Star Trek, or a host of other games, without a computer to store the logic, the responses and histories of the player's performance in multiple games. (An aside: One could play the games by manually executing the logic, but that would be an onerous task beyond the patience of most normal people.) Here the computer system is a central requirement for control of whiz bang hardware and logical progress of the game algorithm. Without a speedy and intelligent little "Maxwell's demon" to control the flow of electrons, you would be unable to play these games at all.

A computer system is a central requirement for the control of whiz bang hardware and the logical progress of a game algorithm.

A Short and Incomplete Encyclopedia of Applications

The application of a technology such as computer systems by individuals depends upon price reductions to the point where people can afford the systems without going bankrupt. The first major breakthrough in this area was provided by our hosts today, MITS Inc, with the Altair 8800 introduced a mere 15 months ago. To quote the marketing blurbs, now that the "age of the affordable computer" has arrived, individual applications are possible. Here is a short and incomplete encyclopedia of contemporary applications ideas, the inputs to the prognostication process.

Relieving Onerous Tasks

Here the emphasis is primarily upon the "busywork eliminator" aspect of computer systems; however, in many cases additional functions are added to the basic task to make the result a more comprehensive solution to the problem.

April 15 comes *in* a few short weeks. Have you ever considered the prospect of an automated tax preparation process? The minimal automation is that of bookkeeping and records coupled with the calculational capabilities of your computer. More elaborate aids to recovering as much money as possible is the use of your computer to model the various ways of combining deductions and options such as income averaging so that the tax you pay is reduced to the minimum within the currently applicable rules. (You can also pull off a bit of "cybercrud" intimidation the next time the auditor calls: "Well, sir, my computer is programmed according to your rules, so it must be right.")

One of my major problems is keeping track of my record library. I like to listen to classical music of the 18th, 19th and occasionally the 20th centuries. I have a record shelf which is heavily burdened with my collection, and no way (outside of imperfect human memory) at present to tell whether I already have a record or not when I am in a record store. As a result, my collection has several unfortunate duplications. An eventual application for my own home computer system will be the generation of a personal record catalog which I can bring with me when I go to record stores for a buying spree. The work involved in setting up a file card version of the system is so large that I'll never do it; but using my computer to keep track of the library, I can automatically generate an updated list after each trip to the record stores around Boston.

How many times have you thought about the problem of mailing lists? If you are involved in a computer club's newsletter operation, the problem is probably at the forefront of your consciousness whenever the newsletter is mailed; but lists are useful for a number of personal purposes as well. Do you partake in the sending of greeting cards which occurs each year? If you do, automation of the list of card recipients will greatly improve the time efficiency of that operation (although some purists might say it lacks a certain "personal" touch). Mailing lists and files of commonly used addresses

Continued on page 90



You're the captain of a crusading starship against the logic of your "8008" or "8080". Your mission: search-and-destroy a random number of alien ships, without running out of time, out of fuel, out of ammunition or out of the galaxy. Your galaxy consists of 64 quadrants, in which there are 64 sectors. You must plan your mission to destroy all aliens. But, every time you move you lose a stardate and precious fuel. Don't run into a roaming star that could damage your ship! And, don't forget how much fuel your warp factor uses! Suddenly, *Condition RED! Alien in sight!* But, you don't know how big he is. Fire a phasor or torpedo? He's damaged or destroyed. But, you've used up valuable fuel. Does he fire back? How about the fuel used for your protective shields? Be careful. You're running out of time and fuel! But, don't give up hope. There *are* refueling stations out there. It's your job to maneuver logically, strategically, carefully to complete your mission.

Here's the multidimensional microcomputer game you've asked for. It's got everything you need for exciting intergalactic adventure. A total program in book form in machine language, for 4K memory: flow charts, illustrations, and more. The program gives you a new, different game every time. Order your copy of SCELBI's GALAXY GAME BOOK today. Only $14.95 ppd. Use Master Charge. Then blast off on your own mission in the galaxy.



ONLY 14.95 ppd.

SCELBI COMPUTER CONSULTING INC.

1322 Rear Boston Post Road
Milford, CT 06460 • 203/874-1573

Pricing, specifications, availability subject to change without notice. Prices for U.S. and Canadian delivery at book mailing rate. Add $2.50 for each publication if Priority Air Service (U.S.) desired. Overseas include $5.00 for each publication for Airmail service.



Photo 1: When you first open your KIM-1 box, you see a thick layer of documentation, including a large wall chart of the system's hardware details, an MCS650X Instruction Set Summary card, KIM-1 User Manual, Programming Manual *and* Hardware Manual. *Also shown in this picture is the KIM monitor listing copy which must be requested separately and is a must if you are to take advantage of KIM's subroutines in applications programs.*

A Date with KIM

Richard S Simpson
314 Second Av
Haddon Heights NJ 08035

Here it is! In the November 1975 BYTE, Dan Fylstra reviewed the capabilities of the MOS Technology 6501 microprocessor chip in an article titled "Son of Motorola" (page 56). The article stated that "it will be three to six months before you see (a 6501) designed into a kit. . ." Well, MOS Technology has gone one better and introduced not a kit, but a completely assembled, tested and warranteed microcomputer with a price tag of only $250! Using the 6502 processor chip (a 6501 with an on-chip clock), the microcomputer features 1 K of RAM, 2 K of ROM containing the system executive, a complete audio cassette interface, a serial terminal interface, 15 bidirectional IO lines, a 23 key keypad and a six digit LED display. This completely assembled one board computer has all the programming features of the 6502 at a very competitive price.

If you have been hesitating over buying a microcomputer because of the difficulty of assembly and the fear that it won't work when you're finished, KIM-1 is for you. The only assembly required is to attach six self adhesive plastic feet to the back of the KIM-1 printed circuit board and attach a +5 volt, 1 ampere power supply to the 44 pin edge connector provided. You'll also need a supply of +12 V for the cassette interface; but a handful of flashlight batteries should work fine since only about 50 mA of +12 V is required, and that only when the interface is being used.

The name KIM is an acronym for Keyboard Input Monitor. The name really describes the ROM executive routines, not the whole unit, but it's a pleasant change from the manufacturer's name followed by a number. It's also significant that the system derives its name from its software.

The KIM-1 board can be operated in one of two modes: using the on board keypad and LED display, or using a serial terminal. The keypad and hexadecimal display is infinitely easier and less error prone than throwing toggle switches and reading results from binary lamps. In fact, for program entry and many simple applications, I prefer the 23 key keypad and bright LED display to my slow, noisy Teletype. The keys have a good, positive "feel" to them (MOS Tech-



Photo 2: The KIM-1 processor as it is removed from its box. The MOS Technology product comes in a neat package which has one foam padded and static protected KIM-1 board as its bottom layer.

nology should know about such things, since they are a major manufacturer of chips for calculators).

The switch in the upper right corner of the keypad puts the machine in single instruction (not single cycle) mode. When the switch is "on," each depression of the "GO" button causes a single instruction of your program to be executed. Control is then returned to the executive program in ROM and the contents of all six machine registers (PC, X, Y, S, P, and the accumulator) are stored in fixed memory locations where you can easily examine them through the keypad or terminal and then "GO" to the next instruction. This is an important capability, since if you just halt a microprocessor after each instruction there is no way of examining the registers (they're all inside the chip!).

I won't go into any detail on the instruction set (see Dan Fylstra's article for that) except to say that it is comprehensive. The variety of addressing modes makes complex programming (especially when processing lists) a lot easier. The 6502 architecture has no IO register or IO instructions, so any memory location can become an IO "port" if you build the hardware for it. KIM comes with a built-in 15 line bidirectional IO interface. TTL levels are acceptable, of course, and one of the lines can supply enough current (5 mA) to directly drive a power transistor. The manual shows how to use it to drive a small speaker for "microprocessor music" programmed in a manner similar to the Kluge Harp of October BYTE (page 14). Each line can be separately programmed for input or output by writing a status word into the correct memory location.

The cassette interface is carefully thought out and should be foolproof. Half of the executive ROM is devoted to the cassette interface software, which includes rudimentary file management and sophisticated programmed equivalents to UART operation. This software allows multiple dumps to a single cassette. A header written on each output segment allows you to say, in effect, "find me program number 34 on the tape and load it starting at location. . ." A checksum is stored at the end of each segment and the user is immediately informed if the computed checksum doesn't match when the tape is read back in. You can even record voice data between segments of digital data – the interface will ignore the voice. This feature could be used to verbally record the instructions for a game and then automatically load and run it. Both high and low level outputs are provided to interface with any type of cassette recorder. It's not a vital feature, but it indicates the care with which the entire system has been thought out.

The TTY interface is for a standard 20 mA current loop (figure 1 shows how I modified it for an RS-232 interface). A unique feature of the software is automatic data rate detection. As soon as the system is powered up, the user types a RUBOUT character on his terminal. The software

If you have been hesitating over buying a microcomputer because of the difficulty of assembly and the fear that it won't work when you're finished, then KIM-1 is for you.

KIM-1 derives its name from the software, a significant indication of the importance of good user support programs.

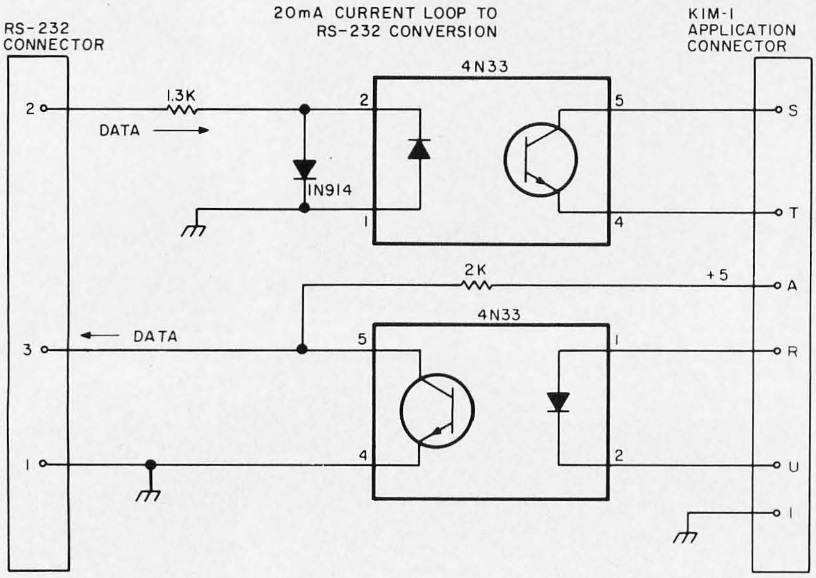


Figure 1: One way to interface KIM-1 with an RS-232 compatible terminal is illustrated in this diagram. Opto isolators are used to accomplish the coupling. The RS-232 pins 1, 2 and 3 will be sufficient for terminals which do not involve handshaking; on some terminals, pins 5, 6, 8 and 20 of the standard RS-232 plug may have to be tied together to bypass handshaking signals.

calculates the data rate (anything from 110 to 1200 baud is acceptable) and automatically adjusts all further conversation to that rate. No additional timing standards or switches are required for the interface.

The real beauty of the terminal interface is in the software, not the hardware. On request, MOS Technology supplies a complete listing of KIM. All the executive ROM software subroutines are documented and available to the user referencing this well-commented listing. Thus, to print the contents of the accumulator in hex on the terminal requires a simple one-instruction subroutine call. Those readers who have had to invent their own terminal interface software will have a deep appreciation for this capability. Similar subroutines are provided for reading characters from the terminal or keypad, printing one or a string of ASCII characters, or writing digits in the LED display.

To round out the terminal interface, software is provided in ROM to read and punch paper tape if your terminal is so equipped. Again, care has been taken to provide checksums on the punched tape which is automatically verified when the data is reloaded. This kind of attention to detail reflects the high caliber of the MOS Technology offering. One reason for this is the fact that MOS Technology sells a sizeable portion of the KIM units to industrial users. This policy of building to industrial rather than consumer standards is also evident in the quality of the PC board, the PC artwork, and the fact that the board is coated with a solder mask, a plastic coating which protects the printed wiring. To further emphasize their faith in KIM, MOS Technology gives you a 90 day warranty on the entire KIM system, not just the components. Mail-in repair service is available even after the warranty expires.

Interval Timer

Another feature of KIM which is finding its way into more and more microprocessors is the inclusion of a program controlled interval timer. The KIM board actually contains two programmable timers, but one is dedicated to control the keypad and cassette interface. Any count from 1 to 256 can be loaded into the timer by writing to the timer's memory location. The user can control the scale of the timer by programming it to count every clock pulse or to count every 8th, 64th, or 256th clock pulse. This prescaling of the counter is done by decoding the last two address bits for the timer. Thus, the time scale is controlled by which memory location is loaded with the count. You might consider using a similar scheme whenever you have to write more than eight bits to control an external device: Just use the least significant address bits as data.

When the timer has counted down to zero, a software interrupt is generated, notifying the program that "time has run out." As soon as the interrupt is issued, the timer continues to count past zero (into negative numbers) at the clock rate. If the program is servicing other interrupts, it can read the counter register to determine how long ago (in machine cycles) the timer interrupt occurred.

Memory Expansion

If you are interested in expanding the KIM memory beyond the 1 K provided, you'll be glad to know that all the decoding for the first 4 K is provided right on the KIM board. All you need to provide is 4 K more of RAM chips and some buffers.

There are two connectors on the KIM board; one called the expansion connector is for adding memory and bus oriented devices. The second connector, called the application connector, interfaces directly to the outside world. The expansion connector has all the address, data, and memory control signals. The application connector terminates the lines for the audio cassette, the terminal send and receive signals, and the 15 IO lines. Connections are also provided so that the keypad can be removed from the KIM board and mounted elsewhere, a useful feature if

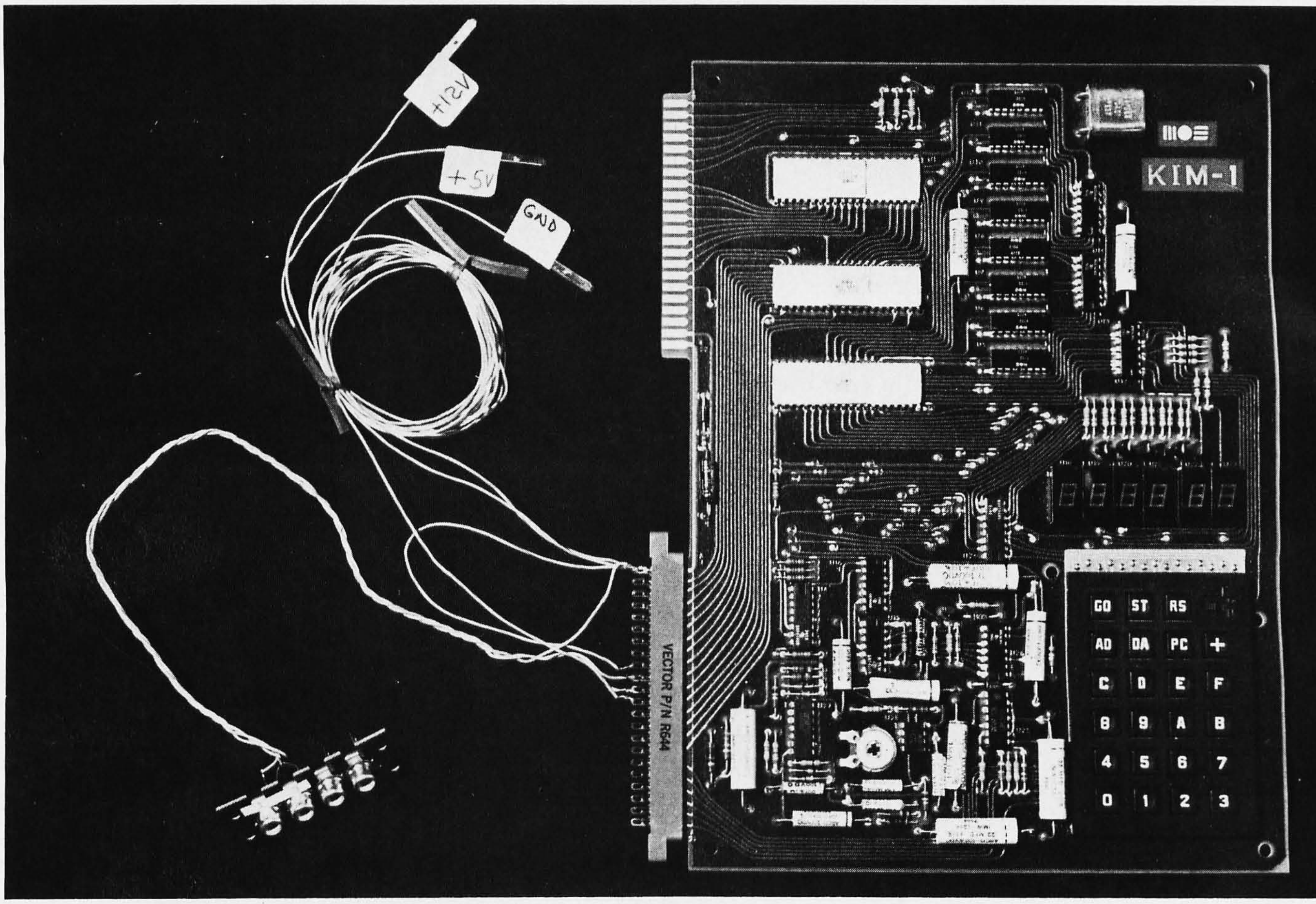


Photo 3: Wiring for Stand Alone Use. With due respect to the instructions in the KIM-1 user's manual, and addition of some miscellaneous parts, the results will be a wiring harness similar to that shown here. Wires have been attached and labelled for GND, +5 volts and +12 V. The audio cassette interface has been brought out to an RCA-style phono jack assembly purchased at a retail electronics store, along with interconnection cables for the recorder input and output. This setup enables the user to enter and test out programs through the KIM-1 control panel and LED display.

you want to wrap up the KIM printed circuit board in sheet metal along with a power supply.

Documentation

The documentation which comes with KIM is thorough and comprehensive. Any regular reader of BYTE should have no trouble following the details of the 200 page programming manual. There are plenty of examples; and the explanation of the operations which occur in each machine cycle of multicycle instructions, while not essential, is very instructive. Special sections of the manual are devoted to interrupt handling and use of the stack pointer. This is vital information often glossed over in other manuals.

I have to admit that I have not yet digested all the information in the 150 page hardware manual which came with my KIM, since my main interest is in programming my system as soon as possible. However, the manual seems to have a solid emphasis on IO interfacing and usage of the control lines.

The third manual provided is the actual KIM user's manual. This 100 page document explains how the keypad, cassette interface and terminal interface are to be used. It gives a few basic programming examples, including an example which goes through the entire design of a simple application using the IO lines. My only complaint is that no sample program was provided for the use of the programmable timer or the ROM executive subroutines. Also, the listing of KIM should have been supplied as a standard item.

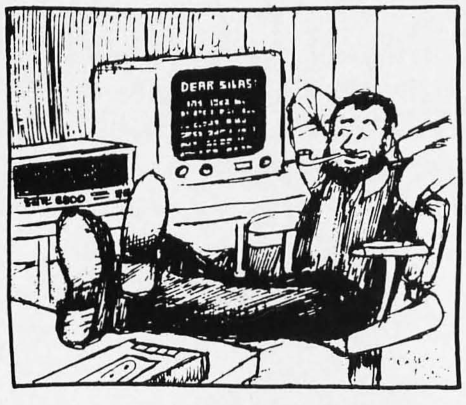
Also included in the package is a pocket reference card for the instruction set and a wall size schematic of the entire KIM board. Two other useful documents are available from MOS Technology on request. One is the manual for the 6500 cross-assembler, which is available on several commercial time-sharing systems. The other is the well-commented listing of the executive programs stored in ROM as mentioned earlier.

In summary, the KIM is an excellent microcomputer requiring no assembly and which is very attractively priced. The only auxiliary equipment required is a power supply and a cassette recorder. The manuals are among the best available and the built-in keypad and display make KIM easy to get started with. The terminal interface and ease of memory expansion make it easy to upgrade as your requirements increase. Make a date with KIM – you'll enjoy it! ■

Letters

Establishing BYTE Committees of Correspondence

To encourage correspondence among readers, beginning with letters received after May 1 1976, BYTE will print the name and full address of each published letter's author. If you do not wish your address to be printed, mark it "do not print my full address" or the logical equivalent.



ON THE TRUTH AND BEAUTY OF BLINKING LIGHTS (AND OTHER SUBJECTS)

The "Total Kitchen Information System" was a big hit since I'm constantly pestered by people who want to know what I could possibly do with a computer in my abode. Well done!

I am glad to see that BYTE is attempting to reverse that great movement to rid computers of blinking lights. A computer isn't a computer without blinking lights! Just compare an IBM 370/158 to a 370/155 (which it replaced) to see what I mean. Or how about the six foot light panel of a 360/195? Lights Forever!

There seems to be a good deal of interest in the game of Space War. I wrote a version of Space War for an Adage AGT-40 graphics system a couple of years ago that was pretty successful. Based on my experiences, I could not guarantee that the game will be suitable for running on current micros since the computation overhead is fairly high. However, some BYTE readers should be able to make simplifications to the game which will permit some level of it to be played.

Kevin Kelley
Wappingers Falls NY

ON CODE TRANSLATIONS AND VACUUMS

In your editorial in December 1975 BYTE you expounded on the need for a common high level language to facilitate the exchange of software between different computer systems. Another approach to the problem might be to write some sort of translation or cross-assembler routine to convert, say, 8080 into the equivalent, say, 6800 instructions. Such a scheme would use less memory than a high level language compiler and therefore be of more use to users with small systems (or budgets). I'm not sure how valid the scheme is and thought that perhaps you or BYTE's readers could determine its validity.

I would also like to join Mr Ryland's lament on the software vacuum.

Another gripe along these lines is the "literature vacuum." The manufacturers supply basic information on their microprocessor and that's about it. Take, for example, the 8008. It was around for years with only Intel's information until Martin Research came out with *Microcomputer Design,* a virtual encyclopedia on the 8008. It's a fantastic book, and I can't seem to find such an "encyclopedia" for any other microprocessor. This makes it tough to compare microprocessors without buying a system based on that microprocessor, which gets expensive.

Anyway, BYTE's pretty good so far – keep up the good work.

Brian Greiner
Deep River, Ontario

Where architectures are similar, translation between instruction sets is quite feasible. Whenever assembly code takes advantage of "special characteristics" with no direct equivalent in the target machine of the translation, the result of a simple translator will be what could at best be termed "inefficient" code. A complicated translator which takes advantage of special cases would tend to eat up a lot of memory for its program, just as a compiler or high level language does. One of the slowest methods of all would be to implement an interpretive simulation program on the target machine, which can execute the instructions of the source machine program. Such simulations are typically 20 to 50 times slower than real time execution on the source machine.

These comments are obviously not the last word on the subject.

A QUERY ABOUT THE AUDIO STANDARD

I read with interest BYTE's proposed cassette standard. I'm in agreement with all the specifications but one: the choice of mark and space frequencies. I think the mark frequency should be lower than the space frequency.

It is desirable to be able to read into the computer two or more blocks of data as one "file." For instance: To assemble a program on a tape that was produced by dumping several TVT pages – to do this without error requires that the entire interblock gap appear to the computer as "mark." To do this without wasting tape requires that the drive be stopped in the interblock gap.

As the standard is proposed, special circuitry is required to "edit" the tape stoppage. If the mark and space frequencies are

The Enlightened Altair.

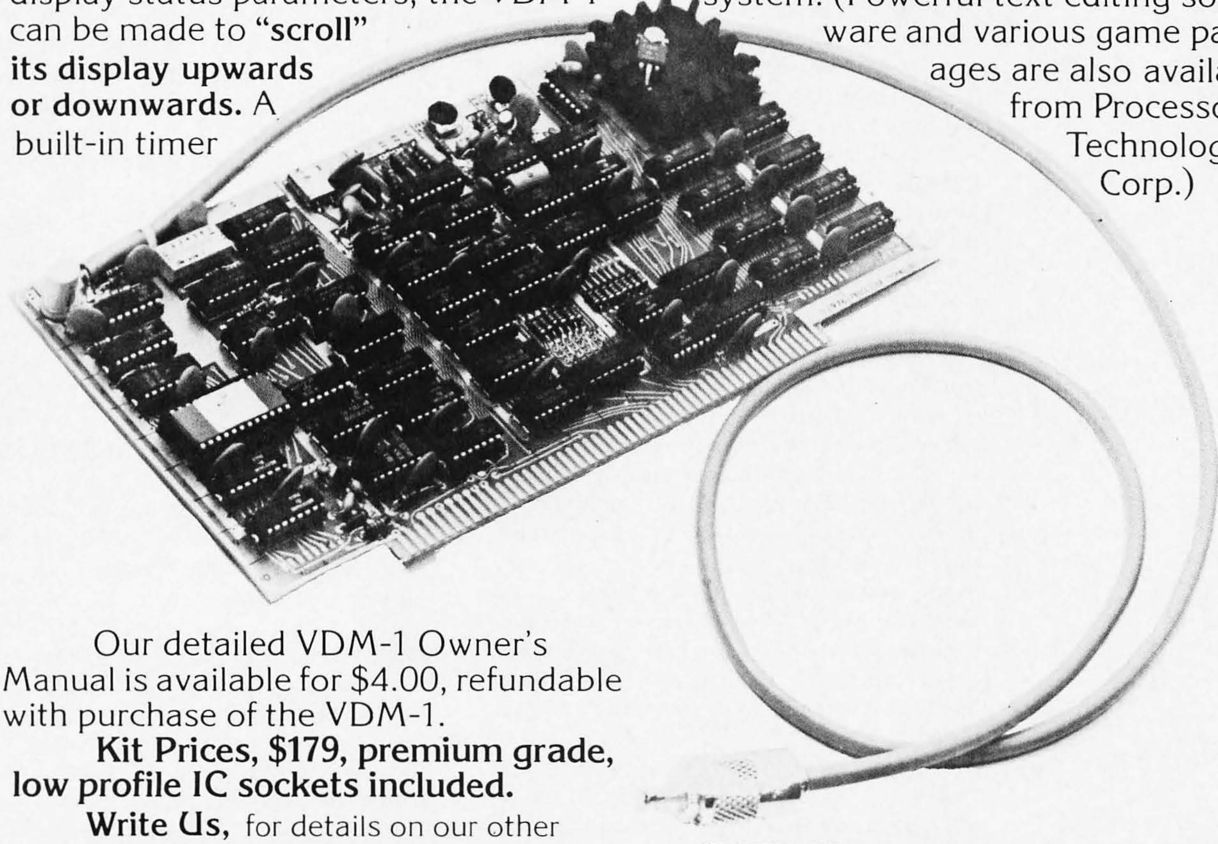
Your Altair already has the intelligence, so let our **VDM-1 Display Module** make the best of its capacity to communicate. This is **not** a limited "TV Typewriter." The VDM-1 is an **ultra-high speed output device, built right into your 8800 system.**

The VDM-1 generates **sixteen 64-character** lines in a large easy-to-read font with both **upper and lower case letters.** It contains 1K (1024) bytes of random access memory, to which the processor can read or write, just as though the memory were an integral part of the system. As the information is written in, contents of the on-card memory are displayed instantly without interrupting the operation of the processor.

Once the processor provides the display status parameters, the VDM-1 can be made to **"scroll" its display upwards or downwards.** A built-in timer allows scrolling at about 4 lines per second, eliminating complicated timing program routines. At top speed, the display scrolls through a dump of 65K of memory in two minutes; that's about 1000 lines per minute!

Multiple programmable cursor circuitry is built in. All 1024 cursors can be displayed at one time or begin anywhere in the display. Thus, the VDM-1 can display white-on-black or black-on-white—perfect for many video games! The VDM-1 also features EIA Video output for any standard video monitor, or a television repair shop can easily modify your own TV set.

The VDM-1 comes with **free terminal mode software,** designed for teletype replacement when used with BASIC or our own Resident Assembly system. (Powerful text editing software and various game packages are also available from Processor Technology Corp.)



Our detailed VDM-1 Owner's Manual is available for $4.00, refundable with purchase of the VDM-1.

Kit Prices, $179, premium grade, low profile IC sockets included.

Write Us, for details on our other compatible 8800 plug-in modules.

Processor Technology

6200-Y Hollis Street
Emeryville, CA 94608

exchanged, and no writing is done while the tape is not at speed, no added circuitry is needed.

Michael W Fellinger
Boulder CO

Harold Mauch, one of the participants at the standards conference, replies:

Mr. Fellinger raises several interesting points which were considered by the participants in the BYTE symposium. I think significance depends on the manner in which the cassette standard is implemented and used. I believe Mr Fellinger is assuming the interblock gap produces the same kind of deciphered output as the "marking" tone produces. This is not necessarily the case. First of all the interblock gap created by stopping and starting the cassette is full of trash caused by de-energizing and re-energizing the record head while the tape is decelerating and accelerating. This is heard on playback as a "chirp." Even if the mark and space frequencies were interchanged, the problem remains since the lower marking frequency will "chirp" to the space frequency and momentarily be interpreted as a space. Second, there is a precedent in data communication for interpreting the absence of signal as a space. This is the idea behind a "break." It is useful because it notifies the user or equipment of a signal or line fault. The teleprinter runs "open" or the "break" indicator comes on.

The important point is the need for proper operating procedure to prevent the "gap trash" from producing errors.

Strictly speaking, either of the two frequencies could have been chosen as the "marking" state and would have made very little difference if the states were deciphered with a phase locked loop or other FM type discriminator. The type of cassette player with which this standard will be used "rolls off" or attenuates the higher frequencies. Consequently the higher frequency tone will be somewhat lower amplitude during playback than the lower frequency tone. Since the signal level is most conveniently adjusted during the "marking" interval preceding a block of data, it is desirable that the marking tone be the lowest amplitude of the two tones. This would be the high frequency in most cassette players. The lower frequency tone would then always have a somewhat greater amplitude than the reference adjustment level increasing the immunity to signal dropout.

Choosing the higher frequency as the "marking" state also permits circuit economies if the deciphering is done digitally.

A recorded character is "framed" by a leading "space" bit and trailing "mark" bits. The beginning of a character is denoted by the mark-to-space transition at the beginning of the start bit. Obviously it is desirable to identify this event as precisely as possible. In the circuit described in the March issue of BYTE (and subsequent revision in the April issue and previously by Don Lancaster in BYTE No. 1) the existence of the lower frequency is indicated by the full cycling of a retriggerable monostable. The fact that it is allowed to time out indicates the presence of the lower frequency immediately and unambiguously. Consequently the high-to-low frequency transition produces a relatively precise event. On the other hand the low-to-high frequency produces a condition somewhat like saying "if you don't hear from me I'm not going." This leaves open the time interval in which to make a decision. Specifying that interval and acting on it involves a slight circuit complication which is not necessary with the proposal as stated.

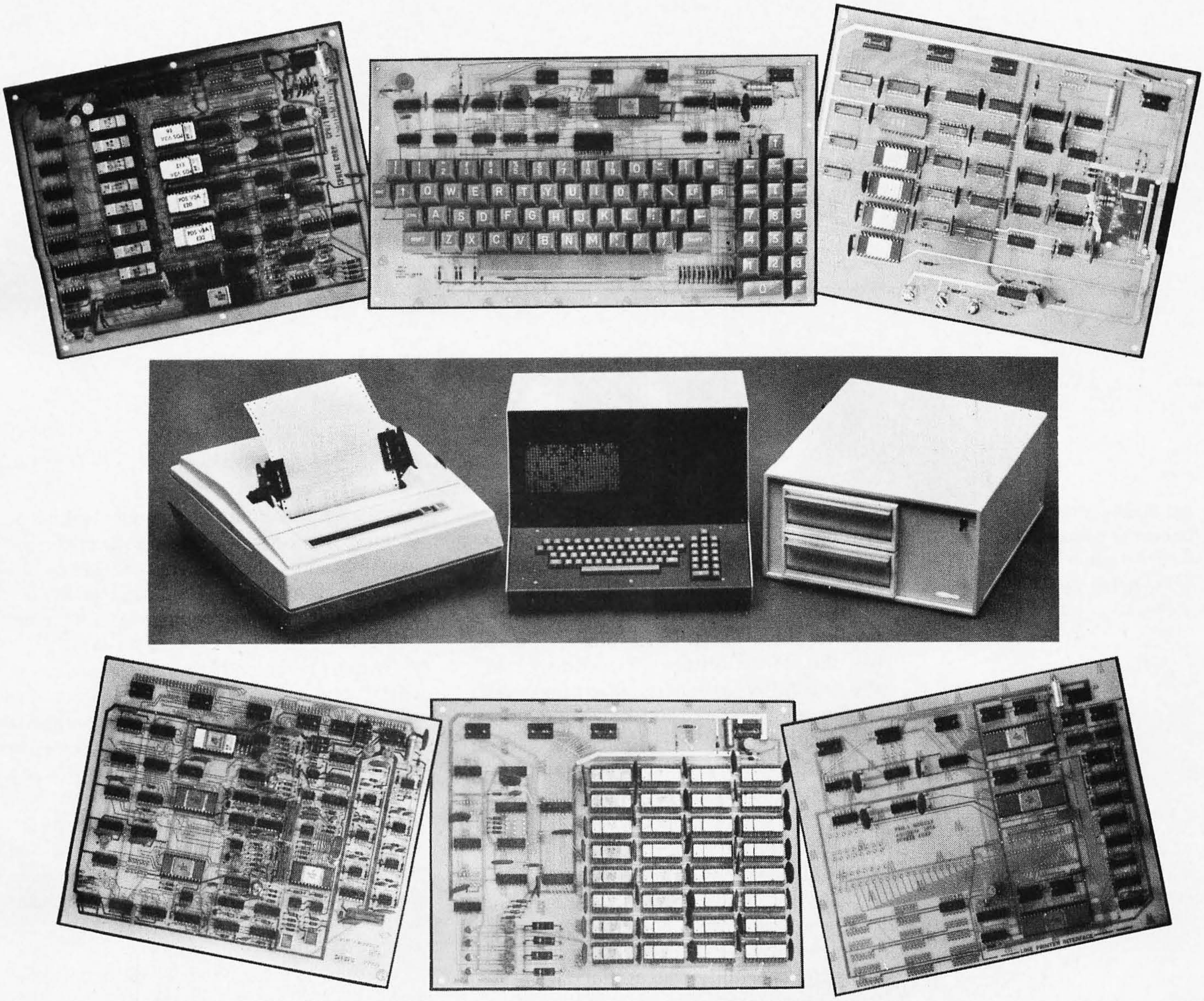
All of the above comments aside, the important point is the need for proper operating procedure to prevent the "gap trash" from producing errors. When operating manually do not permit the computer to utilize the cassette output until well into the five second "marking" leader preceding each block of data. Identify the end of each block with a special character. For example: a line feed or ETX code if the content of the block is text or an asterisk (*) if the content is a program. This tells the computer to inhibit further response to the cassette. When operating automatically (computer controlling the tape unit remote control input) have the control program wait for a second or two of "clean" marking interval before accepting data from the cassette. If the "no signal" condition produced the same output as the "marking" state, this procedure could not be implemented as simply and effectively as it is.

Harold A Mauch
Dallas TX

PAYING OHMAGE TO RESISTANCE

The article on standard abbreviations, "K or k?" in the January 1976 BYTE by Manfred Peshka was interesting, but I noticed one major flaw in the abbreviation used for the unit of resistance, the ohm. Using the letter O is a very bad idea as, when it follows a number, it is difficult to identify the number and the units. For example, the article on blinking lights on page 53 had the following line: "The 222 O resistors . . ." I first thought this to be a typographical error until I read the standards article. It is always a bad idea to use O for anything when using the number 0 at the same time

Continued on page 80



n Source

RD Boudinot PhD
Computer Sciences Corp
6565 Arlington Blvd
Falls Church VA 22046

For integrated circuits, physical and electrical equivalents sometimes have identical designations — and sometimes have completely different numbers.

The practice of one company supplying an assembly or a component which is equivalent to a product of another company is called second sourcing. In fact, there are often many alternatives to the original supplier; thus the title of this article. We shall discuss the history of second sourcing, why second sourcing has flourished, what the buyer's risks are, and how to approach the decision process within the second source environment with specific application to personal microcomputer systems.

The Price Umbrella

The financial basis for second sourcing is the nature of the marketing strategies that have been historically applied to computer systems. Pricing has been a game of balancing the capability of a product against its manufacturing cost. The strategy has been to produce a series of systems where each system is more capable than the one below. The low end machine usually sells at a small profit. By designing each system for eventual upgrade, it is possible to double the power of the entry level machine with a disproportionate hardware cost. For example, twice the capability may be expected to cost the user slightly less than twice as much. If the increased capability is achieved by simply changing a CPU clock, the profit potential is obvious. The difference between a Burroughs 2500 and 3500 is an example of a board-change upgrade.

Figure 1 depicts a predicted end user price versus capability using an arbitrary 1.8 factor. In this figure, 16 times the capability costs 10.49 as much to the user. It seems like a bargain, but manufacturing 16 times the power may have only cost twice as much. At this point, the vendor would be yielding 500 percent profit on the upper end machine and only a narrow margin on the lower end machine. The stage is set for second sourcing by companies who offer "twice the capabilities at half the price."

It must be noted that many other factors, costs, and risks, affect a manufacturer's pricing. His real decision is a function of market research of how many machines he might expect to sell, how much it will cost to design, fabricate, and market them, how many will be upgraded initially or later, etc. All costs for machine design are spread over some number of machines (usually the number of high probability projected sales) along with fabrication costs and markup. Then the final prices are established.

If sales are as good as or better than predicted, fat profits may result. However, a large risk is always present, as the demise of the computer production divisions of GE and RCA has shown.

History of Second Sourcing

Second sourcing has been with the electronics industry for a long time. It has long been expected that several manufacturers would make a 10 uF, 16 V tubular capacitor or a 1000 O, 0.25 W resistor. The same applies to tubes and transistors. One may purchase a 6BE6 manufactured by GE, RCA, or Sylvania, among others. Likewise a 2N2222 may be purchased from Motorola, RCA or Texas Instruments. Finally, among discrete components, electrical equivalency is often cross-referenced between a given manufacturer's line of transistors and other transistors not manufactured by him. A common tactic is to make a general purpose

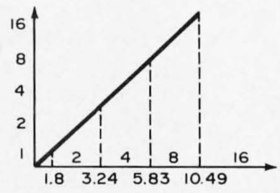


Figure 1: End user cost versus capability. End user cost is shown on horizontal axis for increasing capability on the vertical axis. In the example, increasing the cost by an arbitrary factor of 1.8 doubles capability.

device which meets or exceeds the requirements of a large number of devices. The Radio Shack line of 37 transistors which cross reference to 20,000 other designations is an example. The first lesson many people learn about second sources in discrete components is that electrical equivalence does not guarantee physical equivalence. Attempting to fit an equivalent part with a TO-3 case into a circuit board where a failed TO-92 case was originally used teaches a lesson which is rarely forgotten.

The advent of the integrated circuit brought greater attention to second sourcing. The 9000 series and 7400 series pioneered by Texas Instruments and Fairchild were quickly copied, once success was evident, by numerous competitors. ICs, like discretes, became fair game; confusing equivalent numbers were eliminated and everyone now numbers a 7410 as a 7410.

The S1998 clock chip manufactured by American Microsystems, Inc (AMI) is an electrical and physical equivalent of the MM5316 made by National Semiconductor Corporation. However, the 1103A made by AMI is an equivalent of the Intel 1103A 1 Kb RAM. In LSI (large scale integration) second sources sometimes are numbered the same as their equivalent, and sometimes not.

Large Computer Second Sourcing

The potential profitability of second sourcing was described above. Such profits can be realized only in a successful market. Just as IBM became the giant controlling more large computer sales than all its competitors, likewise, it became the obvious and somewhat vulnerable target for second sourcing. Although large computers and systems existed for a decade, it was not until the late 1960s that viable second source alternatives were developed and marketed with significant success.

IBM received substantial competition from vendors offering: (1) peripherals, such as disk drives and controllers and remote terminal controllers, (2) core memory, and (3) terminals. Several law suits followed, the most famous of which was Telex versus IBM and IBM's subsequent countersuit. Telex accused IBM of monopolizing, and IBM accused Telex of stealing proprietary information.

The efforts to divert IBM business continue from a variety of sources. In all areas of peripherals, core, and terminals, alternatives to IBM equipment exist; in some cases there are many choices. As with components, alternate devices sometimes bear designations which disclose relationships and sometimes they do not. For example, an Itel 7330 disk system replaces an IBM 3330 disk system. On the other hand, an alternative to an IBM 3270 CRT is the ADDS 980A CRT.

Minicomputer Second Sourcing

Second sourcing in the minicomputer environment has been aimed primarily at peripherals and core. Sophisticated terminal subsystems are not generally used with minicomputers and certainly not in the quantity to make the area as lucrative as it is in the large scale computer environment. Among peripherals, alternate sources are often derived simply by modification of an interface. If a peripheral maker who supplies brand X designs a tape reader and punch or line printer whose interface is easily modified, then supporting brand Y with the same

Second sourcing is universal for discrete components — generally any 100 O resistor will do, and in many circumstances any 0.01 uF capacitor will suffice in an appropriate application.

Increased production with a minimum of additional overhead means more competitive prices and reduced manufacturing costs.

device requires only a small capital investment. Increased production with a minimum of additional overhead means more competitive prices and reduced manufacturing costs.

The second source alternatives in minicomputer core memory developed as a result of two factors. First, the pricing strategies applied to large scale computers were also applied to minicomputers. Therefore, each successful minicomputer created a second source market. Prime examples are the Data General Nova 800 or 1200 and the Digital Equipment Corporation (DEC) PDP-8 or PDP-11. Other less widely sold systems (in terms of total sales) such as the Hewlett Packard HP-21XX series, have also been the object of second sourcing when a large amount of expensive, additional core is added to a computer for the resultant increase in productivity. This type of machine generally supports multiple users in a timesharing mode, where the number of users efficiently handled is a function of the amount of core memory available.

Second, advances in solid state memory provided alternatives to traditional core memory. Benefits included cost and speed, although these parameters gradually improved from marginally to substantially in favor of solid state memory.

Microcomputer Second Sourcing

The second source market in microcomputers is directed at memory and IO boards. From all indications, terminals may soon be included in this market. Currently, the peripherals market is limited because tapes, disks, readers, and punches cost more than microcomputers. The only inexpensive devices for external storage now available are the audio cassette type. The peripherals market can be expected to expand rapidly, as will the alternate sources. Microprocessor users will continue to appreciate the significance of the chip name: Microprocessor. Computers are "processors," which require inputs and produce outputs. Inputs are primarily originated from terminals or peripherals and outputs must be passed to terminals or peripherals. A microprocessor alone has few uses.

For two reasons, main memory is currently the major second source item in the microprocessor area. First, if RAM is purchased from the supplier of the microcomputer kit, 8 KB of memory costs more than the basic kit in almost every case. Second, after terminals, main memory is second in importance to an operative system. The memory available limits the size of programs and data buffers and determines the level at which programming must be accomplished, i.e., machine language, assembler, interpreter, or compiler.

IO control boards represent a unique area of second sourcing. Almost no such parallel exists in large scale computers. Some special purpose boards are available for minicomputers, but they are mostly limited to special AD, DA, or multiplexer interfaces rather than conventional IO interfaces.

Risks to the Second Source User

We have discussed the wide variety of second sources available, ranging from discrete components to peripherals, microprocessors, and IO boards. The risks and problems are just as divergent.

Discrete components such as resistors and capacitors are intrinsically low in risk. They either work or they do not. Their performance parameters are readily measured and little confusion is possible. Likewise, more complex components such as tubes and transistors are easily checked for almost all parameters in conventional testers.

More complex devices such as TTL and CMOS chips are fairly easily tested in terms of switching function, voltages, and loads only at low speeds. Failures at circuit speeds are not easily detected on a unit basis. However, suspected devices are relatively easy to replace, and they seldom fail.

Computer peripherals represent the potential for enormous problems. Some classic fingerpointing contests have centered around systems consisting of an IBM mainframe with brand A tapes, brand B disks, brand C core, and brand D terminals. Isolating the real problem in the midst of five different vendors claiming their hardware meets their specs (yet the "system" won't run), could exceed anyone's patience.

Isolating the real problem in the midst of five different vendors claiming their hardware meets their specs (yet the "system" won't run), could exceed anyone's patience.

Core memory followed a slightly different pattern and the risks have changed. Initially, some mainframe vendors disclaimed any warranties and refused to provide maintenance service if another vendor's core memory was utilized. The courts did not concur, and mainframe vendors were forced to allow second source memory to be integrated with their systems. One problem was thus relieved, but the fingerpointing possibility still exists. Additional microprocessor systems' considerations will be discussed in the next section.

IO and other special purpose boards are susceptible to the considerations discussed above. Additional considerations relevant to their selection are discussed in the next section.

It is clear that some of the problems in using second sources in large scale or minicomputer systems do not equally apply to microprocessor systems. Larger systems are generally maintained by their vendor(s). Microprocessor systems tend to be maintained by their builder/integrator.

How to Approach the Decision Process

The following discussion will address microprocessor systems only. The five considerations in second source selection are:

- Support
- Cost
- Performance
- Physical characteristics
- Electrical requirements.

Each will be discussed separately in terms of its relationships to the decision process. A methodology for deriving a selection that leaves the weighting factors up to the individual will be presented.

The support area includes:

- Initial documentation
- Services available from the factory or regional offices, such as consultation when a problem arises
- Other integral hardware and software.

Documentation includes not only adequate construction instructions but also debug instructions for use if the unit is inoperative after construction is complete or if it fails later. Detailed instructions regarding the use of all switches, in the case of the mainframes, and software tips, in the case of peripherals, are also important. Determining the adequacy of support is the most difficult of all the evaluations to be made. One rule of thumb is to rely on more than one source of information if at all possible and never to use second hand information.

The product cost is the easiest to determine. For investments of significant size, a phone call to check the latest price is often very profitable. Unlike most market areas, microprocessor and related components are continually decreasing in cost. The November issue of BYTE lists the cost of a 4 KB memory board for the ALTAIR 8800 at $264. A conversation with an ALTAIR representative in early November 1975 placed the price under $200.

Performance characteristics may not be readily available. When buying a stereo amplifier, one of the parameters always compared is RMS power in watts. In the microprocessor environment, all timing data is not always published with the advertisement. RAM memory varies almost on order of magnitude in speed. Among the 2102 RAM series alone a 6:1 relationship exists. The speeds of 2102 devices are listed in table 1.

Physical characteristics include height, width, depth, and weight. If a board won't physically fit in an enclosure, electrical compatibility has little meaning (unless you are prepared to rewire by hand). Also to be double checked are plug and socket compatibility and heat dissipation.

Electrical characteristics are very important considerations. The primary parameters are current requirements and the impacts on the bus. Each microprocessor kit that includes a power supply in the cabinet has an inherent limiting factor. Arbitrarily, we will assume that a 10 A, 5 V supply is

Table 1: 2102 RAM access speeds.

Device Designation	Access Time
2102	1000 μs
2102-1	500 μs
2102-2	650 μs
2102-8	1500 μs
2102-A	350 μs
2102-A-2	250 μs
2102-A-4	450 μs

Table 2: Expandability example.

Components	System A	Components	System B
Basic System	2 A	Basic System	2 A
Two IO Boards	2 A	Two IO Boards	2 A
12 Memory Boards	6 A	6 Memory Boards	6 A
Total Required	10 A	Total Required	10 A
System A expandable to 48 KB memory		System B expandable to 24 KB memory	

Microcomputers are truly an idea whose time has come.

included. If the "bare-bones" processor requires 2 A, and each IO board requires 1 A, and two IO ports are required, then the current load before adding core memory is 4 A. This leaves 6 A available. Now we have a decision point based on present and future memory and IO requirements. If the vendor 4 KB memory board costs $200, and the second source board cost $100, the choice may seem obvious. It is often the case that vendor boards require low current. We'll use 0.5 A as an example. The second source example will be 1 A. If no future requirements for special interfaces are planned, then the decision is a straight forward evaluation of core memory expandability. The expandability problem just described is depicted in table 2. Given the same basic system, configuration A, using vendor supplied boards, is expandable to 48 KB of memory.

Configuration B is only expandable to 24 KB memory using the available power supply. The decision is further clouded if two more IO boards are contemplated for the future, resulting in the new limitations of table 3. The decision then is based on whether 16 KB of core, the limitation of configuration D, is sufficient for project requirements. The second main electrical characteristic to consider is the number of TTL loads an alternate board will drive. In a bus system allowing expansion up to 20 boards, each board should be able to drive 20 loads. Otherwise, a limitation to expandability is introduced.

Software characteristics include the availability of vendor or second source software. If a high level language is required, any microprocessor system that does not have a self-hosted, high level language available at an appropriate cost is unacceptable. Software considerations are also appropriate for special interfaces and devices. If supporting software is not available, it sometimes proves to be a long tedious job to develop sophisticated handlers. Finally, firmware (software stored in read only memory (ROM)) may be a major consideration. Firmware monitors and assemblers are available for M6800 systems via the vendors and for 8080 systems through alternate sources. Disadvantages of firmware vary slightly depending on whether ROM or erasable programmable read only memory (EROM) is utilized. If a ROM is "burned in," no options exist for easily modifying or patching the firmware. Such a requirement results when a bug, or program error, is detected or when new applications or hardware make modifications desirable. It is hoped that the latter case is more common. One approach worth considering for a system that is expected to grow in the future is to use a firmware monitor and bootstrap loader to load all operational programs, compilers, and interpreters from an external storage device, such as a cassette. There are certainly cases, however, where a firmware assembler or BASIC interpreter is the best choice.

A Decision Matrix Example

A final example, to be used only as an example, is included. Because each selection must be made on an individual basis, the actual companies and parametric data are not included in the example.

Table 3: Expandability example.

Components	System C	Components	System D
Basic System	2 A	Basic System	2 A
4 IO Boards	4 A	4 IO Boards	4 A
8 Memory Boards	4 A	4 Memory Boards	4 A
Total Required	10 A	Total Required	10 A
System C expandable to 32 KB memory		System D expandable to 16 KB memory	

The first step is to establish a method of ranking each contender in each area: support, performance, physical characteristics, electrical requirements, cost, and software. Within each category, the criteria of excellence must be established by using discrete measures where possible. If the scales to be used will range from 1 to 10, then the criteria for achieving each ranking must be established. Then the relative weightings for each characteristic, perhaps 50 percent cost, 40 percent electrical, and 10 percent software, must be established. Finally, minimally acceptable standards must be determined for each category. It is possible for an item to rank worse than the lowest increment by being a physical impossibility. In table 4, items flagged with an "x" are unacceptable. Thus we see that brand B does not meet the minimum electrical specification, brand C exceeds the cost range, brand D will not fit physically into the cabinet, and brand E was unacceptably slow. Further, in this example, support, physical characteristics, and electrical characteristics have a weighting factor of 0.0. They were factors for elimination but not comparison. The decision was binary: Acceptable or unacceptable.

This decision matrix technique is of considerable value, even if one has no confidence in the weighting factors, because it forces a systematic comparison and evaluation of all characteristics of the choices. It has been used by the author on numerous occasions and has dispelled numerous misconceptions. The hard facts lined up against each other can be most revealing.

Table 4: Sample decision matrix. Weighting scale of 1 to 10; 10 is best. X designates unacceptable.

	Support	Performance	Physical Characteristics	Electrical Requirements	Cost	Software	Score	Weighted Score
Brand A	2	2	3	2	8	4	19	5
Brand B				X				
Brand C					X			
Brand D			X					
Brand E		X						
Brand X	2	5	4	3	2	4	20	6.4
Weighting Factor	0	0.4	0	0	0.5	0.1		

Conclusions

In conclusion, there are three key points to be made. (1) Second sourcing is a healthy phenomenon. In fact, because second sources establish product credibility and guarantee users that someone will be able to supply them, many chip makers actually seek a second source. (2) The benefits to the source user are primarily monetary; however, there are hazards, and they must be considered before making a decision. (3) Finally, it took the large scale computer market a decade and the minicomputer market several years to develop second source markets. Using the date of introduction of the ALTAIR 8800 in January 1975 as the date of initial general availability, the microcomputer industry has achieved this maturity within its first year of existence. Microcomputers are truly an idea whose time has come.■

GLOSSARY

Capability: In a broad sense, the capability of a system is a weighted sum of the individual features which make a system easy to use and reliable. In a more specific sense, a particular capability of a system is a point of comparison with equivalent points on other systems. (A general capability might be the ability of a system to accomplish a particular application; a specific capability might be the mass storage medium employed in a system.)

Firmware: In the context of this article, firmware means software which has been written into a read only memory. A second definition of firmware applicable to microprogrammed machines is the set of microprograms required to emulate a specific CPU architecture.

Microprocessor: This means any of the large scale integration (LSI) computer designs currently available and used for inexpensive personal computing systems.

Second sourcing: The practice of "follow the leader" as applied to the electronics and computer technology. One company designs and markets a product; then a second (or third, etc.) company takes the external functional specifications and creates an equivalent circuit which will accomplish the same functions. The second sourcing is often sanctioned by licensing arrangements. The second source company for any given product is always in a following mode, since the product's innovator is by definition the first firm in the marketplace.

Vendor: The commercial term for the source of a product is "vendor."

Read Only Memories in Microcomputer Memory Address Space

Dale Eichbauer
Digitech
PO Box 6838
Grosse Pointe MI 48236

The important advantages of a ROM in microcomputer use are nonvolatility and write protection for whatever data it holds.

In an earlier BYTE (see "Read Only Memory Technology," page 64, December 1975), Don Lancaster introduced the use of read only memories as a tool for design at the hardware level. This application is but one of a multitude of uses for ROMs, especially when you consider a ROM as part of the main memory address space for your computer. The important advantages of a ROM in microcomputer use are nonvolatility and write protection for whatever data it holds. It relieves the user from the chore of reentering frequently used programs each time his machine is fired up or after data is accidentally modified. To put it simply, your data is *always* in the machine whenever you need it.

A bootstrap or absolute loader is a simple program which just transfers data from an input device to memory. To keep it in your machine, it should ideally be in ROM.

The two most common and well known uses of ROMs are for holding loaders and system programs. There are two basic types of loaders: the bootstrap (or absolute) and the more complex relocating loaders. The bootstrap or absolute loader is a short program which is used to load the machine following a power interruption or any other type of catastrophic failure which wipes out the main programmable memory. (Unless your machine's programmable memory is of a special design, it is volatile, meaning that its data is lost if power to the memory is lost for more than a very short time.) This loader program requests input from a peripheral device such as a paper tape reader or cassette drive which contains programs needed for machine operation and stores this input data in programmable memory. After toggling all your data in from the front panel following power interruption, one can easily see both the convenience and versatility of such a bootstrap loader. The relocating loader takes the input data from the peripheral device, converts its addresses from a relocatable form into absolute binary and stores it in memory at selected addresses. It might typically perform some error checking and turn over execution to the loaded object program.

System monitors are prime targets for ROM technology.

Monitors and Debuggers

System programs suitable for or, preferably, in ROM include such things as system monitors, assemblers, device drivers, software debugging programs, hardware fault testing and diagnostics. The system monitor (which is often available from the computer or CPU manufacturer) is a program which handles and coordinates machine operations at a basic level. A monitor allows the user to control the entire system's operation with simple, powerful commands. A typical monitor might have commands for the creation, modification, and deletion of files, device independent IO (from the user's point of view), automatic assembly and execution of programs, relocation of programs and data, and so forth. Device drivers (short programs which handle the software end of peripheral interfacing) are rarely changed once debugged and are needed for almost all IO operations, making them a natural for ROM storage. Software debugging programs, often manufacturer supplied, provide a means of detecting and correcting programming faults. The many forms and features which they possess are too extensive for any detail in this article. One rather unusual but potentially useful application of ROM storage is in storing hardware testing and diagnostic routines. Testing of the microcomputer often can be done by simple programs which execute an algorithm and compare the results with the correct answer. It can also be done by complex programs which execute all functions of the machine, often in certain critical combinations peculiar to the

machine under test. At first it would seem that there is no need to put these routines in memory of any type until needed except for convenience, since it would be an infrequently used task. Consider, however, the case where a fault which is to be located is in some way related to or impeding the input or the programmable memory's storage functions. If this is the case, then the testing or diagnostic routine may never get into the machine in usable form to do its job.

Simulation and Emulation

Simulation is another use of ROMs in microcomputers which will become more common as CPU capabilities increase, machines proliferate, and users demand more of their machines. Simulation is the technique of interpretively executing an instruction set for one computer design using a program running on a second "host" machine. For example, a host machine with an 8080 CPU could execute object programs from another machine which uses a 6800 or PACE CPU (or even IBM 360/370 software for those with delusions of grandeur). A ROM could contain the simulator program to execute the foreign instruction set. With an appropriate general purpose simulator program it might even be possible to change the instruction set of a machine by referencing a different ROM data table for each simulated machine. Of course all such simulations run much more slowly than the actual speed of the computer in question.

A related technique is emulation, in which microprogrammed hardware implements an instruction set directly. Some microprocessors are internally microprogrammed, but the user typically will not see this fact externally. Microprogrammed computers are fairly widespread in contemporary technology. And with nearly every microprogrammed computer, there is a control store implemented in some form of ROM. But the majority of microprocessor chips currently available do not give the user a facility to use microprogramming techniques. The instruction set is typically committed by the manufacturer during the design stage; so, to perform the software of a foreign machine, a software simulator must be used as described above.

With such simulations, the slowness of operation is due to the fact that a series of instructions (a subroutine) must be executed on the host computer in order to achieve the effect of a single instruction of the simulated machine. Even though a simulated computer may be 10 to 50 times slower than the real machine, such slowness is often tolerable when compared to the time it would take to hand translate the program. Use of ROM to store the simulator makes the simulation mode virtually a part of your hardware, protected from destruction due to power loss or accidental modification during program execution.

Subroutines

Another excellent use of ROMs is the storage of subroutines. Multiply, divide, double precision, floating point, conversion formulas and other algorithms, plus additional software implemented functions are in the machine as soon as power is applied. When they have been implemented in ROM, such subroutines act as if they were really hardware instructions.

Security Data

Anyone assembling a multi user computer system, especially one with remote access, should consider using a ROM for maintaining data pertinent to the various users of the system. This data might include such things as access codes, what devices and memory segments are authorized for use by which individuals, the particular user's system priorities (for job and device scheduling by the operating system), and so forth. The operating system constantly needs such information to make decisions concerning the handling of tasks for the current users. A ROM protects this information from modification or destruction, whether accidental or malicious.

Tables

An excellent use for ROMs is the storage of tables of values. There are many tables, such as logarithmic, sine, cosine, and tangent values, which could be of use to almost any computer hobbyist. A program needing one of these values then has to merely look up the desired value in the appropriate ROM table. Such tables can also be used to speed up high precision calculations by giving an approximate starting value. Those faced with interfacing a non-ASCII encoded terminal or other peripheral (such as EBCDIC, Selectric, Baudot, or Hollerith) to their microcomputer may find that a character conversion table, implemented in ROM, is part of the solution, as Don Lancaster points out in BYTE #4. However, while his conversion scheme uses a ROM which does its conversion of data apparently at the peripheral itself, in many cases it would be useful or desirable to perform this conversion in the machine. Such a conversion method would even make it possible for two terminals, whatever their coding scheme, to commu-

If you plan to do a lot of simulation, the simulator program might be a logical choice for ROM. With diligent software preparation, your humble 8008 could simulate a mighty 360/370 (although much much more slowly in execution).

A library of often used subroutines is another item which would make a good candidate for ROM storage.

Data tables for character code conversion via software can be stored in ROM if they are used a lot.

nicate with each other using the microcomputer (and its ROM) as a sophisticated interpreter. And, if data rates, character lengths, and line lengths are different, then such a setup offers the added advantage of using software and memory as a buffer to compensate for these differences.

Waveforms

If your machine is equipped with a DA converter (digital to analog converter), then a ROM can contain a set of values which, when output through the DA, will produce a custom waveform. In many cases special waveforms may be generated in this fashion which would be impractical to generate, using any other method. Both the frequency and amplitude of the waveform may be controlled completely by software. With an 8 bit word and a DA with 10 volts full scale output, resolution of 0.04 volts per bit is obtainable. The maximum generated frequency is dependent on the speed of the microcomputer and the number of outputs per cycle required for a suitable waveshape.

Error Checking and Arithmetic

Two other possible uses for ROMs which may be implemented either in main memory or as processor add-ons are a parity generator/checker and a fast multiplier/divider. A table for all possible combinations of a word can be referenced to generate the parity bit or a flag check bit. Multiplication and division may also be done as table functions. Several of the IC fast multipliers currently available are actually modified and specially programmed ROMs.

The article in BYTE #4 also introduced Programmable Read Only Memories (PROMs), which are the most useful type of ROM for computer hobbyists, since a custom pattern costs very little to have programmed or the user can do it himself.■

If you want to use your computer as a low frequency (audio) waveform generator, you could burn a set of standard waveform patterns into ROMs, using software to drive a DA conversion device at various frequencies.

Bibliography on ROMs and PROMs

These articles are found in engineering publications, which should be available in well stocked corporate or university libraries.

"PROMpting a minicomputer" by Robert Hightower of Motorola in the February, 1973, *Electronic Engineer/Systems Engineering Today*. This is a description of a bootstrap (or absolute) and a relocating loader for a PDP-11 which is stored in ROM.

"PROMs, Proms, Promises" by Jerry Metzger in June 16, 1975, *Electronics Products Magazine*. This is a good introductory article on PROMs and includes a wall chart of all PROMs available, both bipolar and MOS, as of its publication.

"PROMs – a practical alternative to random logic" by Dave Uimari of Signetics in the January 21, 1974, *Electronic Products Magazine*. Here is an excellent article on PROM theory and use which also includes lengthy discussions on programming, such as how it is done, best place to have it done, typical large and small scale equipment, etc.; lists PROM programming services and equipment manufacturers.

"Designer's Guide to Semiconductor Memories – Part 1" by Robert J Frankenberg of Hewlett-Packard Data Systems in August 5, 1975, *EDN* magazine. This is a good introduction to all types of memories, ROMs and PROMs included; it also includes an excellent list of references.

"Read-Only-Memories in computers – where are they headed?" by Roger R Dussine of Compagnie Honeywell Bull and Robert M Zieve of Honeywell Information Systems in the August 1, 1972, *EDN* magazine. The authors provide an overall survey of ROMs, their use in computers, mentions use for fault location, bootstrap, some unusual types of ROMs, and things to come in ROM technology.

"Programmable ROMs offer a digital approach to waveform synthesis" by Karl Huehne of Motorola in the August 1, 1972, *EDN* magazine. This is a detailed description of ROM waveform synthesis.

"Large Bipolar ROMs and p/ROMs Revolutionize Logic and System Design" by Joe McDowell of Monolithic Memories, Inc in the June, 1974, *Computer Design*. Here you'll find a short survey of the current bipolar ROM technology and some examples of use, including a ROM controlled timing pulse generator under microcomputer command.

"Mixing Memories in Minicomputer-based Control Systems" by Richard A Farwell of Data General in the February, 1973, *Control Engineering*. This is a discussion of how various memories are used in Data General minicomputers and the costs and tradeoffs involved; a section on ROMs lists a number of uses outlined in this article.

Manufacturer's data sheets on particular devices contain a wealth of information and are free for the asking. As an example, the data sheets below contain listings of ROM and PROM lookup tables of values.

From AMI:

- A 256 word sine and cosine table in the S8614 supplemental note.
- An arctan table in the S8771 supplemental note.
- A 512 word sine and cosine table in the S8772 data sheet.
- A Hollerith to USASCII conversion table in the S8457 data sheet.
- A USASCII to Hollerith conversion table in the S8539 data sheet.

From Nitron:

- A Hollerith to ASCII conversion table in the NCM 1112 data sheet.
- A Selectric to ASCII to Selectric conversion table in the NCM 1151 data sheet.
- A 512 word sine and cosine table in the NCM 1141 data sheet.

From Computer Microtechnology:

- ASCII to EBCDIC and EBCDIC to ASCII conversion tables in the CM 2850 supplemental note.



We doubt it.

When it comes to microcomputers, Altair from MITS is the leader in the field.

The Altair 8800 is now backed by a complete selection of plug-in compatible boards. Included are a variety of the most advanced memory and interface boards, PROM board, vector interrupt, real time clock, and prototype board.

Altair 8800 peripherals include a revolutionary, low-cost floppy disk system, Teletype,™ line printer, and soon-to-be-announced CRT terminal.

Software for the Altair 8800 includes an assembler, text editor, monitor, debug, BASIC, Extended BASIC, and a Disk Operating System. And this software is **not just icing on the cake**—it has received industry wide acclaim for its efficiency and revolutionary features.

But MITS hasn't stopped with the Altair 8800. There is also the Altair 680—complete with memory and selectable interface—built around the new 6800 microprocessor chip. And soon-to-be-announced are the Altair 8800a and the Altair 8800b.

MITS doesn't stop with just supplying hardware and software, either. Every Altair owner is automatically a member of the Altair Users Group through which he has access to the substantial Altair software library. Every Altair owner is informed of up-to-date developments via a free subscription to **Computer Notes.** Every Altair owner is assured that he is dealing with a company that stands firmly behind its products.

After all, we didn't become the leader by messing around. Shouldn't you send for more information or visit one of our Altair dealers?

Altair Coupon

Please send me the following information:

☐ Your latest catalog and price list
☐ Software information package
☐ Please include a list of your dealers

NAME____________________

ADDRESS____________________

CITY____________ STATE & ZIP________

mits
2450 Alamo S.E. Albuquerque, N.M. 87106

More Information on PROMs

Roger L Smith
4502 E Nancy Ln
Phoenix AZ 85040

Have you ever wanted to program your own read only memories automatically so that you could copy programs into a permanent storage device? This article concerns one kind of erasable read only memory, the Intel 1702A integrated circuit and its pin compatible equivalents the National MM5202AQ and MM5203Q. These memories store 256 eight bit bytes of data using a method which allows total erasure and reprogramming many times. The method of programming is complex while erasure can be accomplished simply by exposure to an ionizing radiation (such as ultraviolet light). When you need to store large tables of data or programs, use of such read only memories is a very attractive alternative to more elaborate types of memory provided a method of programming is available. These erasable read only memories are economical as well, since typical prices at the time of this article are in the $20 range.

Why PROMs?

A few years ago, it became apparent that the different users of read only memories (ROMs) had many special applications which required only one or two copies of any given data pattern. The technology of *mask programmed* read only memories is only cost effective for large production runs of parts so an alternative had to be found. A means was needed for the user of read only memories to inexpensively field program one or two copies of a data pattern. This is where Harris Semiconductor, a division of Harris Intertype Co., entered the picture and coined the term PROM for programmable read only memory, a Harris trademark that has become almost generic through widespread use. A PROM then was simply a ROM that could be programmed in the field. While production read only memories are manufactured from specific masks provided weeks in advance by the user, a PROM can be programmed in seconds automatically by the user reducing turn-around time to a minimum.

Types of PROMs

Let's examine some of the different PROMs in use today. There are a number of options for the memory elements used in making programmable read only memories including nichrome fuse links, diode matrices, stored charge devices, amorphous semiconductors, polycrystalline silicon fuses, etc. Note that all these memory elements can be electrically altered in order to store data. A few can also be restored to the original condition; these are used in erasable read only memories (EROMs).

Figure 1 illustrates how the basic PROM operates. The first thing to notice is a decode circuit. This decodes the address to select one of the 32, 64 (or whatever) word gates in the memory matrix. The decoder is simply an array of multiple input gates with one input for each address bit and one gate for each memory word.

Each decoder gate drives a multiple emitter word driver transistor. In series with each emitter is a memory element which in this case is a fusible link. In this example, we have a 4 bit word so each word driver transistor contains 4 emitters, each connected to a fusible memory element. The memory elements then connect to the appropriate bit sensors and output buffers (4 in this example).

When a particular word is addressed, its decoder and word driver transistor turn on. If the fuse link is intact, the bit sensor turns on and the output line for that bit goes low (logical zero). If the fuse link is open, the

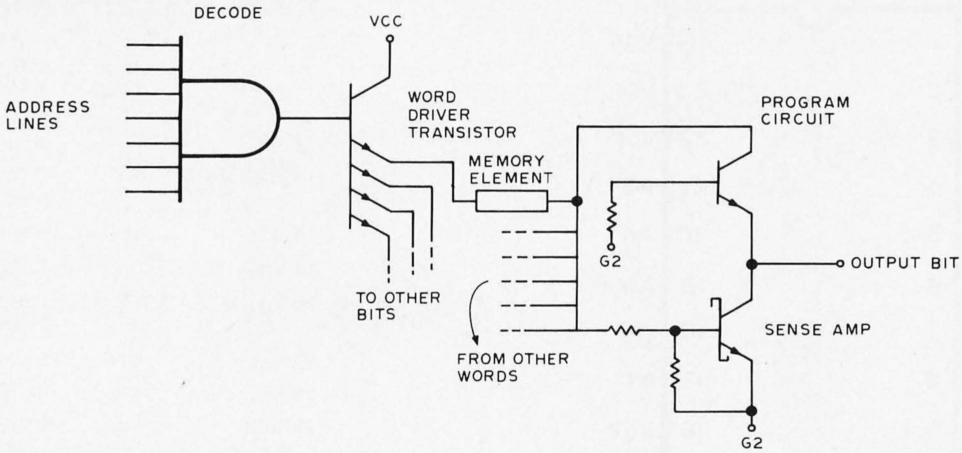


Figure 1: This partial schematic of a PROM shows the circuit for one word and one bit. This PROM would be the nichrome fuse link type.

sensor and buffer circuit remains off and the output is high (1 bit).

Not shown in the diagram are the chip select (or chip enable) lines. The chip select lines are typically connected to the higher order address bits. When many PROMs are utilized, an external decoder circuit (such as 74154 or 7442) might be used to decode several high order address bits and decide which PROMs to enable or select. Essentially, the chip select inputs are used to turn on the output bit sensors and buffers when the PROM is selected. PROMs use open collector or tri-state output buffers so that they can be bused. The buffers are in the high impedance state until enabled.

The nichrome fusible link type of programmable read only memory is manufactured by Harris, Signetics, Texas Instruments, and Motorola. From this basic nichrome fuse PROM, other types have evolved. The next natural step was to polycrystalline silicon fuses, as made by Intel and Advanced Micro Devices. These are easier to build in the semiconductor fabrication process because the fuse links are also made out of a semiconductor material. The silicon fuses are burned open in the same manner as the nichrome fusible link type. Due to the semiconductor structure of the memory elements, these PROMs often require a more elaborate programmer than the nichrome fuse type.

Another development in memory elements is the Avalanche Induced Migration (AIM) device patented by Intersil. Fabrication of these elements is similar to TTL logic which simplifies the manufacturing process. The elements are basically NPN transistors arranged in a matrix with common collectors on the X-lines and common emitters on the Y-lines. In programming a logical one, a high current is forced through the desired transistor from emitter to collector. The emitter to base junction is forced beyond normal avalanche and into secondary breakdown. Aluminum flows into the junction causing a base to emitter short that in effect leaves a base to collector diode. These PROMs are programmed using 2.5 us pulses of 200 mA current that are alternated with sense pulses. After a number of pulses, a change is sensed and the programmer moves on to the next bit.

Erasable ROMs

A memory element used by Intel and National Semiconductor is a stored charge type called a FAMOS transistor. FAMOS stands for *floating-gate avalanche-injection* MOS charge-storage device. It is similar to a P-channel silicon gate field-effect transistor with no contact on the gate. Programming the FAMOS type of memory element requires a pulse more negative than −30 volts applied to the drain or source P-N junction. High energy electrons are injected into the floating silicon gate. With this negative charge on the gate, there is current conduction between the source and drain of the FAMOS transistor.

The primary advantage of this stored charge type of memory element is that the charge can be removed later by exposing it to a high intensity, short wavelength ultraviolet light. The radiation creates an ionizing action that causes the charge on the floating gate to leak back to the substrate. These erasable ROMs (EROMs) are provided with a transparent quartz lid to allow exposure to the radiation. More about erasure later.

For the really dedicated computer hobbyist who wants all of his system monitor, resident assembler, text editor, etc. in PROMs because they are all working as desired (at least this week), erasable ROMs

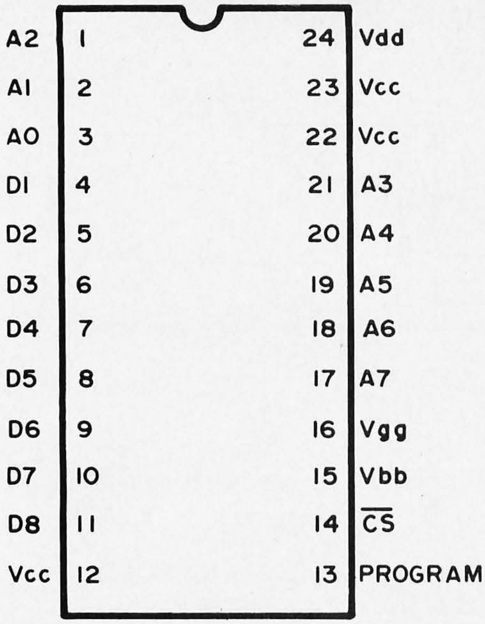


Figure 2: Pin-out diagram Intel 1702A EROM. A0 – A7 = address inputs; D1 – D8 = data output (for READ mode), data input (for PROGRAM mode); $\overline{CS}$ = chip select.

are the logical choice. Currently available for around $20 are the 2 Kb Intel 1702A and National MM5202AQ and MM5203Q. All of these EROMs use the FAMOS stored charge memory elements and can be erased with ultraviolet light. These EROMs have one definite advantage over regular ROMs; they have been tested before delivery.

Intel 1702A EROM

The Intel 1702A EROM is produced in a 24 pin dual in line package with a transparent quartz lid. Intel also makes a 1602A ROM which is identical to the 1702A except that it has a metal lid and is not erasable. All chips undergo complete programming and functional testing on each bit position prior to shipment. The 1702A and 1602A are both 256 word by 8 bit, entirely static MOS ROMs with no clocks required. All inputs and outputs are TTL and DTL compatible, but the outputs are tri-level to allow output busing capability. Memory expansion is simplified by use of a chip select input which disables the chip when high (logical one). Figure 2 shows the Intel 1702A pin connections while table 1 shows the voltage inputs for the read or program modes.

Erasure Methods

To erase EROMs such as the 1702A, Intel recommends using the Model S-52 ultraviolet lamp available from Ultra-Violet Products Inc., San Gabriel CA (cost is about $170) or through Intel distributors. An inexpensive eraser can be built for about $15 using a General Electric ultraviolet lamp #G8T5, a ballast transformer, single pole switch, a push button starter switch, and mounting hardware. The lamp is mounted in an enclosure and the EROM is placed under it at a distance of 0.25 inch. The lamp is turned on for about 6 minutes for complete erasure, but use caution not to expose anyone to the ultraviolet rays.

Table 1: Intel 1702A EROM input voltages.

Pin	Read Mode	Program Mode
12 Vcc	5 V	ground
13 PROGRAM	5 V	Program pulse (–46 V to –48 V)
14 $\overline{CS}$	ground	ground
15 Vbb	5 V	12 V
16 Vgg	–9 V	Pulsed Vgg input (–35 V to –40 V)
22 Vcc	5 V	ground
23 Vcc	5 V	ground
24 Vdd	–9 V	Pulsed Vdd input (–46 V to –48 V)

> CAUTION: When using an ultraviolet lamp, you should exercise extreme care not to expose your eyes or skin to the rays. Short wave ultraviolet light can cause sunburning of the eyes and skin.

According to a National Semiconductor engineer, the ultraviolet erasable EROMs cannot be indefinitely erased and reprogrammed. After about 52 cycles of reprogramming, the device will not work properly unless it is reconditioned by baking in an oven at 400°F for 45 minutes. After reconditioning, the program-erase cycle can be repeated another 52 times, although the National Semiconductor engineer recommends only 35 cycles between reconditioning.

EROM Programming

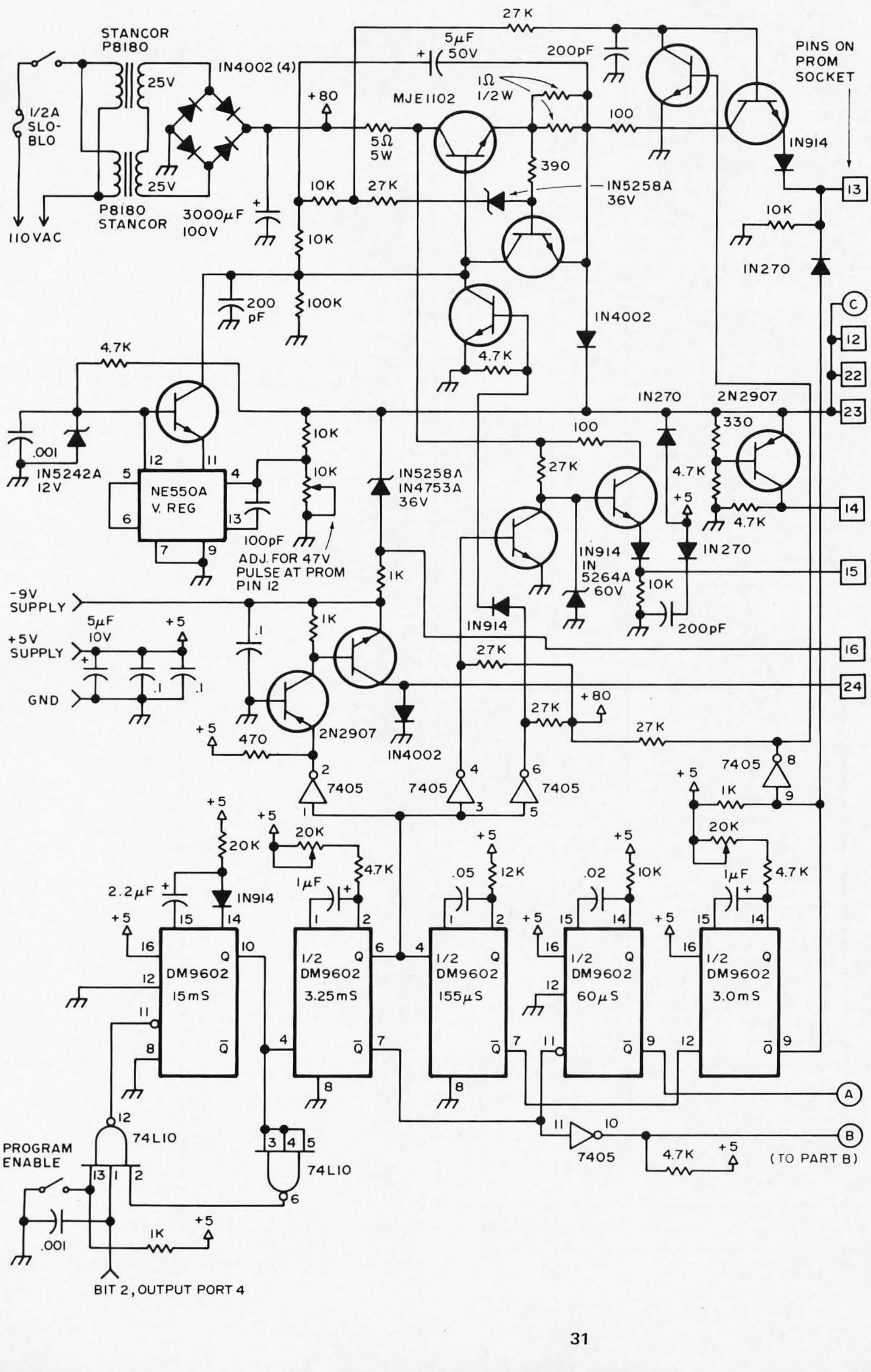
We'll describe two possible methods of programming these EROMs. The first method is *highly* recommended, will prove least expensive, and is extremely simple – order it programmed or send it in to be programmed!

Many EROMs are not simple to program. The 1702A type EROM requires a series (over 32) of 47 volt programming pulses of 3 ms duration with 20 percent duty cycle for each word. Also, at the beginning of each pulse, the address must be complemented.

Manual programming is out, and the cost of an automatic programmer may not be justified. Remember also that in order to erase programs you must buy some type of ultraviolet lamp. A PROM/EROM programmer could, however, prove to be a very interesting and fund raising activity for an industrious computer club.

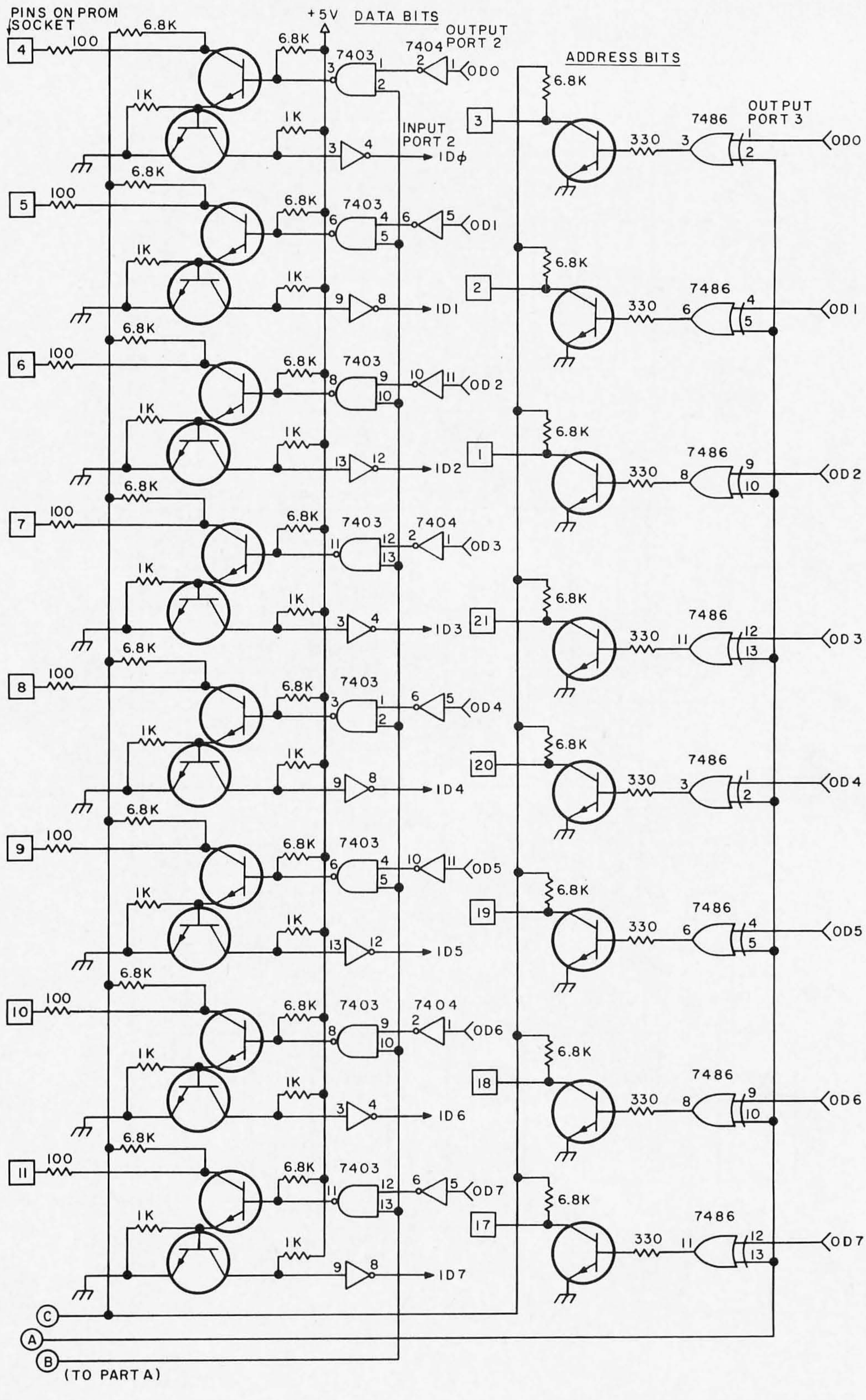
If this isn't convincing enough, or if you plan on going into the business, or if you're just plain curious, you may want to try the circuit of figure 3 that can be built to program the 1702A, 5202AQ, etc. The programmer is a simplification of the Intel MP7-03 programmer and is designed to work with the 8080 program of table 3. Crowbar

Figure 3A:



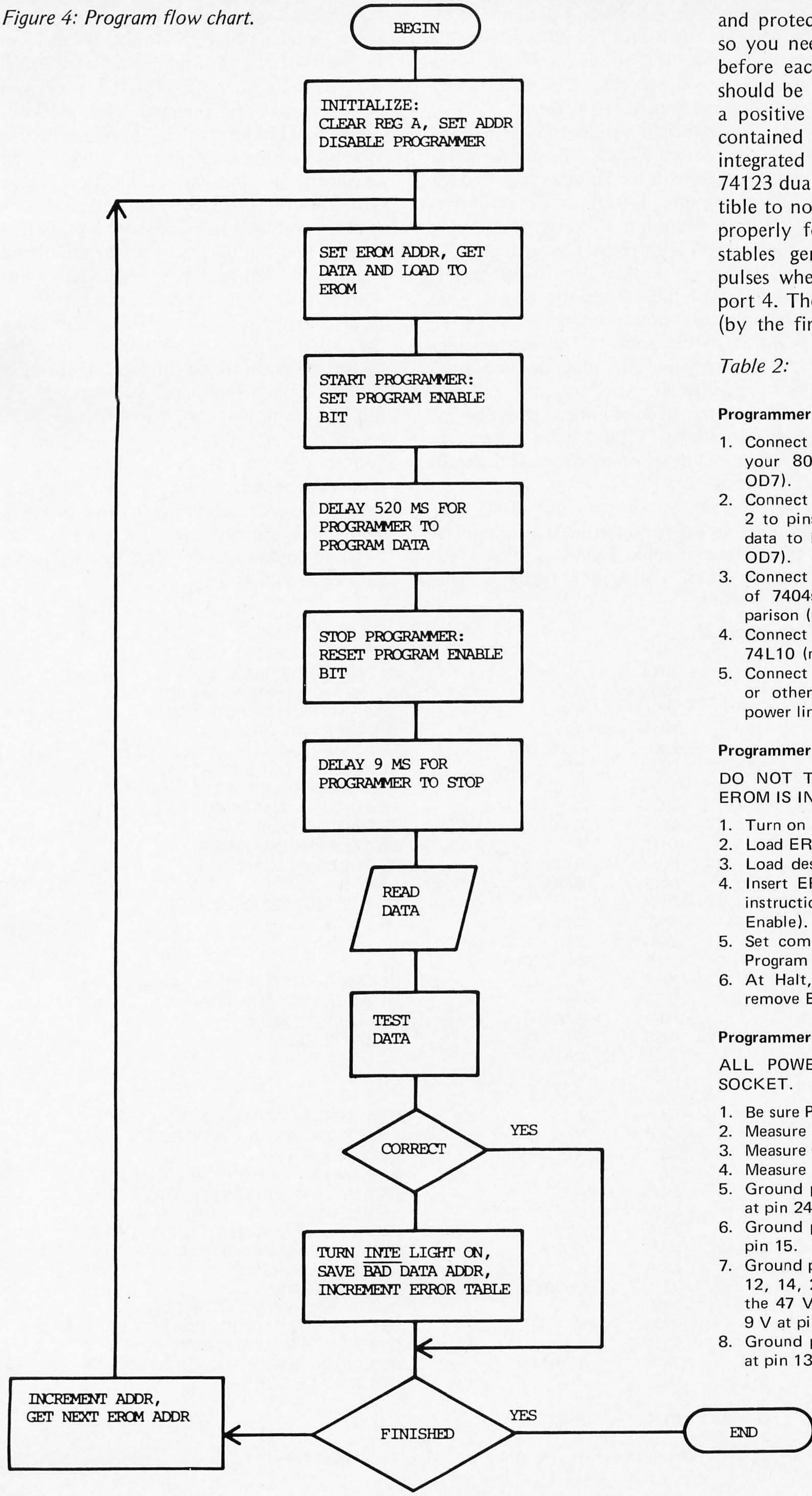
Figures 3A and 3B: Computer controlled PROM programmer for stored charge PROMs. Unless **otherwise noted, transistors** *are MPS-A06 or 2N3722 or equivalent. Pin 14 of ICs to 5 V, pin 7 to ground.*

Figure 3B:



PINS ON PROM SOCKET
+5V
DATA BITS
OUTPUT PORT 2
INPUT PORT 2
ADDRESS BITS
OUTPUT PORT 3
7403
7404
7486
6.8K
1K
100
330
ODO
OD1
OD2
OD3
OD4
OD5
OD6
OD7
IDϕ
ID1
ID2
ID3
ID4
ID5
ID6
ID7
(TO PART A)

Figure 4: Program flow chart.



and protective features have been removed, so you need to check your circuit voltages before each use. Input data and addresses should be in positive logic (i.e., logical one is a positive level). The five monostables are contained in three 9602 dual one shot integrated circuits. Do not substitute the 74123 dual monostable which is very susceptible to noise and therefore may not operate properly for this application. These monostables generate the proper programming pulses when enabled by bit 2 of the output port 4. The pulses are repeated every 15 ms (by the first 9602) and the length of time

Table 2:

Programmer Connections

1. Connect 8 Address lines to output port 3 of your 8080 computer (port 3 lines OD0 to OD7).
2. Connect 8 Output Data lines from output port 2 to pins 1, 5, and 11 of the three 7404s for data to be programmed (port 2 lines OD0 to OD7).
3. Connect computer input port 2 to pins 4, 8, 12 of 7404s for reading EROM data for comparison (port 2 lines ID0 to ID7).
4. Connect bit 2 of output port 4 to pin 1 of 74L10 (near the PROGRAM ENABLE switch).
5. Connect 5 V and –9 V supplies from computer or other source and connect the 110 VAC power line.

Programmer Operation

DO NOT TURN POWER ON OR OFF WHILE EROM IS IN SOCKET.

1. Turn on computer and programmer.
2. Load EROM program at location 001/000.
3. Load desired EROM data at location 002/000.
4. Insert EROM into socket. Single step first five instructions of the program (to disable Program Enable).
5. Set computer at address 001/000, switch on Program Enable switch, and start computer.
6. At Halt, turn off Program Enable switch and remove EROM.

Programmer Calibration Test

ALL POWER ON, AND NO EROM IN THE SOCKET.

1. Be sure Program Enable switch is OFF.
2. Measure 5 V at pins 12, 13, 15, 22, and 23.
3. Measure 0 V at pin 14.
4. Measure –9 V at pins 16 and 24.
5. Ground pin 2 of the 7405 IC and measure 0 V at pin 24 of the EROM socket.
6. Ground pin 4 of 7405 IC and measure 58 V at pin 15.
7. Ground pin 6 of 7405 and measure 47 V at pins 12, 14, 22, and 23. Adjust pot on NE550 for the 47 V. Pin 13 should remain at 5 V. Measure 9 V at pin 16.
8. Ground pins 6 and 8 of 7405 and measure 47 V at pin 13.

each word is programmed is determined by the software. This will be 520 ms with the program given here, so about 35 pulses are generated per word. Note that all bits in a word are programmed at one time.

At first glance it would appear that the voltages generated by this programmer don't seem to agree with the voltages specified for programming the 1702A in the manufacturer's documentation. However, if you look at the voltages with respect to Vcc (pin 12), they do agree. Follow the programming instructions of table 2 exactly and *be sure* you don't turn power on or off while an EROM is in the socket. The monostables could be triggered and program unwanted bits in the EROM. Also, turn the Program Enable switch to off when inserting or removing EROMs. Table 2 gives a complete list of programmer connections and a calibration test.

A program for use in your 8080 type computer to program the 1702A (or equivalent) is shown in table 3 and the accompanying flow chart is shown in figure 4. This program is loaded into your computer starting at octal location 001/000, and the data to be written into the EROM is loaded starting at location 002/000. If any errors in programming are detected, the Interrupt Enable (INTE) light will light and the EROM addresses of the invalid words will be stored consecutively starting at octal location 000/000. The program runs for about 2.25 minutes and then halts. If you have an 8008 system, you can translate the operations and change the timing loops accordingly. For users of other machines, comments and a flow chart are provided to aid in translating the 8080 code for your machine. This EROM program can be entered and operated entirely from front panel switches or, for those with a teletype, punched tape, TV typewriter, etc., the program can be modified to suit. In any event, check that the timing loops are correct for your particular system before attempting to use the programmer to actually program EROMs.

Good luck with the PROM/EROMs for your computer.■

The author will program the 1702A EROM for $5. All he needs is the PROM, an octal or hexadecimal listing of the desired program and $5.

Table 3: 8080 program for EROM programmer.

```
001/000  257                         XRA   A          ;CLEAR REGISTER A
   /001  117                         MOV   C,A        ;SET ERROR TABLE ADDRESS
   /002  137                         MOV   E,A        ;SET FIRST EROM ADDRESS
   /003  323  004                    OUT   4          ;DISABLE PROGRAMMER
   /005  041  000  002               LXI   H,DATBK    ;SET ADDRESS FOR DATA TABLE
   /010  323  003       START:       OUT   3          ;SET EROM ADDRESS
   /012  176                         MOV   A,M        ;GET DATA FROM TABLE
   /013  323  002                    OUT   2          ;LOAD DATA TO EROM
   /015  076  004                    MVI   A,004      ;SET BIT 2 = 1
   /017  323  004                    OUT   4          ;ENABLE PROGRAMMER
   /021  006  260                    MVI   B,260      ;SET DELAY COUNT
   /023  026  000                    MVI   D,000
   /025  025            DELAY1:      DCR   D          ;DELAY FOR 520 MSEC.*
   /026  302  025  001               JNZ   DELA1
   /031  005                         DCR   B
   /032  302  025  001               JNZ   DELAY1
   /035  257                         XRA   A          ;CLEAR REGISTER A
   /036  323  004                    OUT   4          ;DISABLE PROGRAMMER
   /040  006  003                    MVI   B,003      ;SET DELAY COUNT
   /042  025            DELAY2:      DCR   D          ;DELAY 9 MSEC.*
   /043  302  042  001               JNZ   DELA2
   /046  005                         DCR   B
   /047  302  042  001               JNZ   DELA2
   /052  333  002                    IN    2          ;READ PROGRAMMED DATA
   /054  276                         CMP   M          ;CHECK DATA – IS IT CORRECT?
   /055  312  064  001               JZ    ALLOK      ;JUMP IF OK
   /060  373                         EI               ;BAD DATA – TURN INTE LIGHT ON
   /061  173                         MOV   A,E        ;GET DATA ADDRESS FOR DATA
   /062  002                         STAX  B          ;STORE BAD DATA ADDRESS
   /063  014                         INR   C          ;INCREMENT ERROR TABLE ADDRESS
   /064  076  377       ALLOK:       MVI   A,377      ;CHECK IF DONE PROGRAMMING THIS CHIP
   /066  273                         CMP   E          ;AT LAST ADDRESS?
   /067  312  100  001               JZ    STOP       ;HALT IF DONE
   /072  034                         INR   E          ;INCREMENT EROM ADDRESS
   /073  043                         INX   H          ;SET NEXT DATA ADDRESS FOR DATA TABLE
   /074  173                         MOV   A,E        ;GET NEXT EROM ADDRESS
   /075  303  010  001               JMP   START      ;CONTINUE – PROGRAM NEXT WORD
   /100  166            STOP:        HLT              ;HALT – ALL DONE

002/000  (EROM DATA)    DATBK:       BLK   256.
```

*Note: Time delay loops are shown for an ALTAIR 8800 with a clock frequency of 2 MHz. Total loop time = 11.5 usec for an apparent frequency of 1.3 MHz.

Classified Ads Available for Individuals and Clubs

Readers who have equipment, software or other items to buy, sell or swap should send in a clearly typed or printed notice to that effect. The notices are free of charge and will be printed one time only on a space available basis. Insertions should be limited to no more than 100 words. Notices can be accepted from individuals or bona fide computer users' clubs only. Commercial advertisers should contact Virginia Peschke at BYTE for the latest rate card and terms.

WANTED: Back issues of BYTE. I need one each of issues #1, 2, 3, and 4. If you are willing to sell any one of these, please write stating price. K T Matthews, 5959 Student Union Mall, Vancouver BC, Canada V6T 1K2 (604) 224-5672.

SWAP OR SELL: TIL 302 0.27 inch 7 segment LED display, TIL305 7 by 5 matrix LED displays, TIL360 4 digit calculator type displays, 4.255 MHz crystals, regulator boards with +12, +6, and –6V outputs, AA Ni-Cads, Cal-Tex CT5001 and TC5005 calculator chips, Model 26 KSR Teletype, and misc. TTL and linear ICs. Will swap for microprocessors, memories, TVT-II boards, etc. Also need software for Mostek MK5065 microprocessor and would like to contact other 5065 users. Please send address and phone number to C Lee, 1221 Miller Rd, Apt 234, Garland TX 75041.

CORE MEMORY SYSTEM FOR SALE. 16,000 words, 32 bits per word, with schematics. 2 master units and 2 slave units available, also small drum memory unit. Will sell real cheap to best offer I get in cash and or equipment in trade. Call David Kohl, (212) 253-3401, evenings.

FOR SALE: MITS Altair 8800 minicomputer. Factory assembled, $500. Two assembled 4K memory boards, $150 each. Serial interface card 88-SIDA $110. Three 88-EC Extender cards $70. $950 takes all. Call Michael H Chandler, (916) 331-4032 after 5 PM PST.

TRADE OR SELL: One DEC PDP-8/L minicomputer. Has 12 bit word, 4K memory and TTY interface. Will sell for $850 or trade for good optical PT reader. All software included. EDUSYSTEM-10 BASIC, assembler, text editor, tests, etc. Call or write B Wheeler, Box 363, Route 24, Chester NJ 07930. (201) 879-5746.

FOR SALE: MITS 88-SIOA RS-232 serial IO interface for Altair 800 $80 assembled. 88-MCS 256 byte static memory $60 assembled. David Richards, 6655 Hill St, El Cerrito CA 94530, (415) 529-0759 evenings.

SELL OR TRADE: Like-new IBM Model 213 Dictating/Transcribing Machine. 3 inch wide belt and indexing card makes a random-access magnetic storage media; just move the level to the file you want. Far better than a cassette. Complete with remote controls, cover, new tape and a bunch of new index cards! Super neat! Have 6 units. Only $150 each. Also have some Boeing Electronics Call Diverters (demo units). You can use as intended or as auto dialer for computer-controlled applications. Only $225. Call or write R E Larson, 2524 Burnett Ct So, Renton WA 98055 (206) 255-4094.

A DREAM COME TRUE: Two model 35 ASR console Teletypes complete, in superb condition, $1000 each. Also 450 reels of computer tape 2400 feet, most with tape seals, $2 each. Pick up, cash only. Kenneth C Hopper, 4021 S Bowman, Indianapolis, IN 46227, (317) 787-8661.

FOR SALE OR SWAP: 12 sheets of double sided, 1 ounce copper circuit board, 3/32" by 11 1/4" by 16 1/2". $12. Les Harrison, 1201 Country Club Rd, Wilmington NC 28401, 763-5915.

FOR SALE: Two factory made Honeywell H-112 computers. Memory: 12 bit core. Mainframe: Front panel with switches and lights, power supply and cabinet. Software: Editor, Symbolic Assembler, Monitor and Debug, Diagnostics, TIC-TAC-TOE. Schematics and timing specs. Fully functional, need only interface to your peripherals. Build or buy interfaces from Honeywell. ASR-33 + HS Papertape reader interface $225 extra. 4 K CPU (Expandable) $900. 8 K CPU $1200. Terms: Money order and shipping and freight charges collect. Specs only, $10. C Ziemlak, 2 Champlain Dr, Hudson MA 01749.

SMART CRT TERMINAL: Keyboard, two tape drives and controls, color output and power supply. Unit has limited functions with its own processor but your own microprocessor can be worked in. Tape operations include delete, edit, search, etc. Unit, diagrams, two tapes and black and white CRT. $995. Call John at (502) 456-5471, 5 PM to 9 PM EDT.

Attention, Computer Hams. FOR SALE: One Morse code to ASCII (parallel) converter with power supply. Unit connects to speaker output of any receiver. Unit is constructed in attractive wood grain cabinet. Asking $150. Call (607) 754-0309, or write Jack Davis, 3304 1/2 Pearl St, Endwell NY 13760.

FOR SALE OR TRADE: Tektronix 453, excellent condition, looking for disk drive, high speed reader/punch, DECTape, etc, for PDP-8M. Frank Depetrillo, 32 Quincy St, Providence, RI 02908, (401) 751-2374.

FOR SALE OR TRADE: Mitey-Mite TTY w/keyboard IC controller; Drake R4A-TX4 AC3-MS-4-L4 ham station. Want ASR-33. G McKee, 1117 S Rodgers, Okmulgee OK 74447. (918) 756-2978.

WANTED: Model ASR 33 Teletype. Please specify features, condition and price. James E Crane, 5650 Windsor Way #308, Culver City CA 90230, (213) 649-4187.

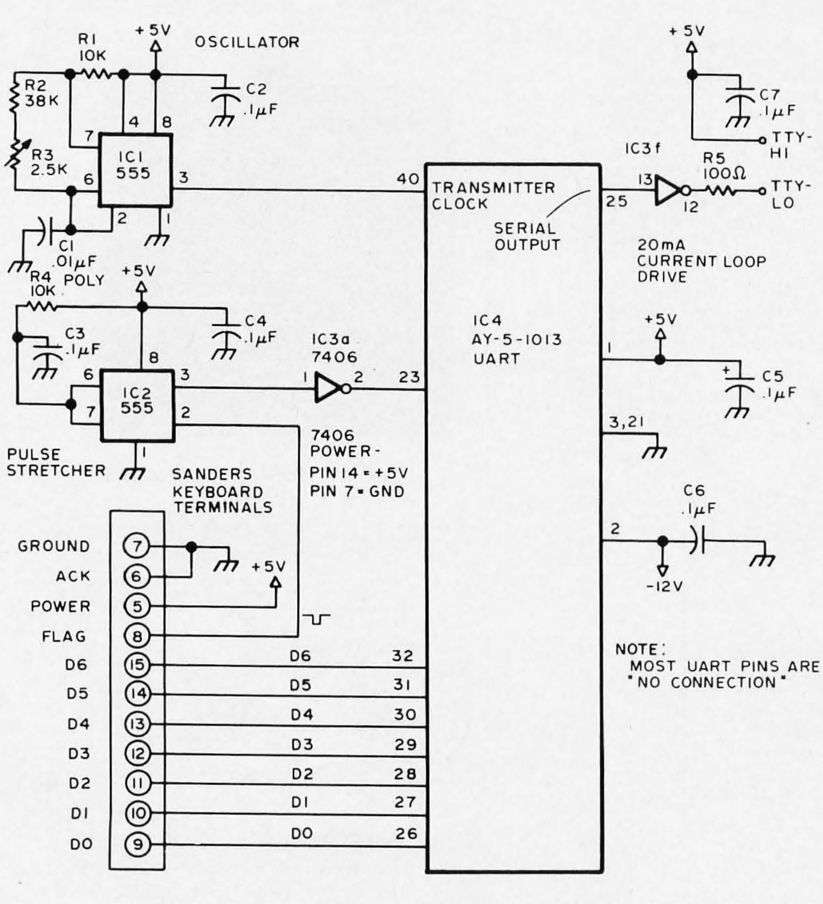
Bits About BYTE Binders

Debbie's in a bind. She can't answer all your binder requests and do everything else too. When we make the binders in the fall, we'll let you know about it in BYTE.■

Serialize Those Bits From Your Mystery Keyboard

Dr George L Haller
1500 Galleon Dr
Naples FL 33940

Figure 1: Parallel ASCII to Serial ASCII Converter. The output of an ASCII keyboard can be converted from parallel to asynchronous serial format using a UART and two 555 timers. The result can be used to drive the 20 mA current loop of the Teletype print mechanism.



Now that you have deciphered your mystery keyboard, (page 62, September 1975 BYTE) and have determined which terminals are for the power supply, data bits, and flag pulse, what are you going to do with it? Well, one good use is to make it part of a Teletype style terminal. The Teletype models 33KSR or ASR, which are complete with printer and mechanical keyboard, are still quite expensive, usually over $1000 new; but the model 33RO, which consists of the printer only, can be bought for less than one half of that price. Now, mate the model 33RO Teletype with your electronic keyboard and you have the equivalent of the 33KSR for your computer terminal. The ASR is the same with the addition of paper tape punch and reader. The computer terminal is usually specified as a full duplex terminal which merely means that while both the printer and the keyboard operate with serial data, they are not connected together except through the computer. The following is a description of a small adapter which will convert your electronic keyboard from a parallel to a serial output device which will then be the keyboard half of your full duplex terminal. The cost of the parts for this adapter, exclusive of power supply, is less than $10.

The main component of this adapter is, of course, the UART which has been used for several years in communication circuits for series to parallel and parallel to series conversion. An excellent explanation of the UART was given in the very first issue of BYTE. (Don Lancaster's "Serial Interface," page 22, September 1975 BYTE.) In order to use the UART, we write in 8 bits of parallel data whenever a key is struck. The key pressed pulse sent to the UART must be negative going and have the correct width to drive the UART strobe. A clock frequency

of 1760 Hz must be applied in order to get a 110 baud data rate out of the UART. The output will produce a high level mark and a low level space. Note that we are only using one half of the UART. The adapter shown here was made for the Sanders keyboard, but it should be applicable to any keyboard if considerations are made to insure that the start pulse sent to the UART is negative going, and data is in true form (logical 1 is a high level). Looking at figure 1, we find that the power is applied to UART pins 1, 2, and 3. The power requirement is about 200 mA at 5 volts (pin 1) and 10 mA at −12 volts (pin 2). The data bits are wired directly from the keyboard to the UART as shown. Terminal 6, the acknowledge function to the keyboard, is grounded. Terminal 8 of the keyboard is the key pressed flag. In the Sanders keyboard, this flag is a negative going pulse which is too short to operate the UART directly. This pulse is first stretched in a 555 timer circuit (IC2). This particular stretcher requires a negative input. After stretching, it is reinverted in a section of the 7406 and applied to the UART. The clock circuit is also a 555 (IC1). The output frequency at pin 3 of IC1 should be adjusted to 1760 Hz. This can be determined by using a frequency counter or by adjusting the potentiometer until good copy is obtained while the keyboard and adapter are connected directly to the 33R0. The frequency should be held to an accuracy of about 1%, but this is no problem with a good polystyrene condenser shown as 0.01 μF. Most of the other terminals on the transmission side of the UART should be a high level input, which means that they can be left unconnected, since they have internal pull ups. The exception is terminal 21 which is grounded. The serial output is connected through the inverter with an external pull up resistor which provides the loop with a mark current of 20 mA and a space current of zero.

Another slight modification of the Sanders keyboard will make it more useful. As received, the keyboard has no "line feed" key. It is a simple matter to convert the TAB key to an LF key. We must change the code for this key from an 013 to an 012 octal, which means we must change the zero bit from a 1 to a 0. Find the terminal at about the center of the rear of the diode matrix labeled "VT". A yellow wire connects this terminal to the TAB key. On top of the matrix board this terminal is connected to a single diode. Either end of this diode should be disconnected. This is the zero bit diode. There are two other diodes still connected under the board which will leave the code 012 octal.■

An Octal Front Panel

Herman DeMonstoy
2 Pioneer Rd
Painted Post NY 14870

A standard method of writing a program into a computer has been to use toggle switches. Being lazy and becoming very tired of toggling in data, I was sure there was a better way. Figure 1 is a block diagram of my "better way" to input front panel data. This is a keyboard control unit which eliminates most of the toggle switches, replacing them with an octal data entry keyboard and various function buttons. No program needs to be entered before using it. The computer sees the data just as if the original toggle switches were there, but one keystroke now performs the function which had required three toggle switches.

Programming with this design is done using octal codes. The reason I chose octal is the fact that a seven segment display does not conveniently decode four bits into hexadecimal digits. I use this design as the main control panel of my new experimental computer based on the 6502 chip.

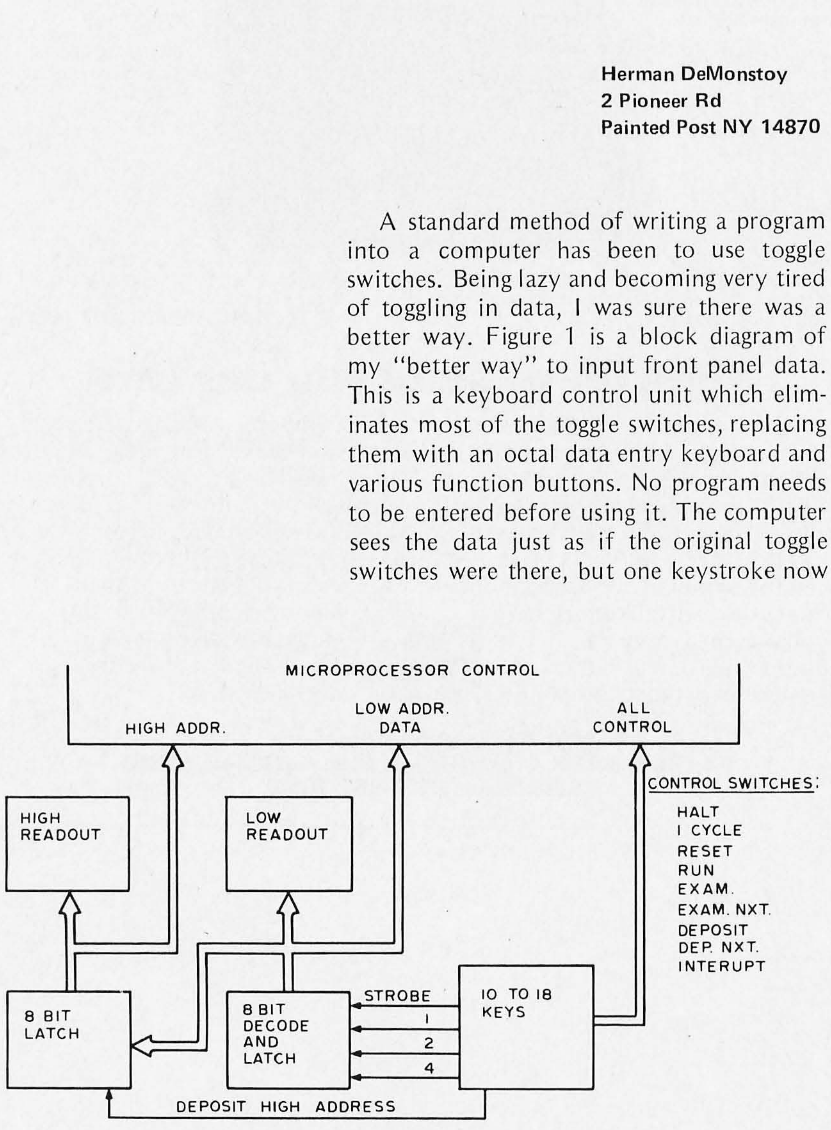


Figure 1: Block Diagram of a Front Panel Control. The idea is to use an octal entry keyboard to enter data which is decoded and latched into several 8 bit registers.

Theory of Operation

Figure 2 shows the schematic of the main portion of the circuit. When a number key is pressed, the strobe line (pin 8 of IC 28) goes high and starts the first oneshot (IC25A). At the same time, the data lines have been set up to the binary code of the key being pressed. The encoding is done by the 7420 NAND gates, IC26A, IC26B and IC27A. The first oneshot writes data into the third latch (IC21) using the old output of the second latch (IC20). At the end of the pulse from the first oneshot, the second oneshot (IC25B) is fired. This writes data from the first latch (IC19) into the second latch (IC20). At the end of this pulse, the third oneshot (IC24A) is fired. This writes the new octal digit into the first latch. In this manner data will shift to the left one octal digit position each time a key is pressed. Progress of the shifting can be watched in the octal display connected to the outputs of the entry latches. When the correct data is found in the display, the deposit switch or other functions associated with the computer can be activated to enter the data into the computer. A second 8 bit latch to accept high order address information is shown in this design. By pressing the DHA (Deposit High Address) key, data is transferred from the entry byte to a second one byte latch

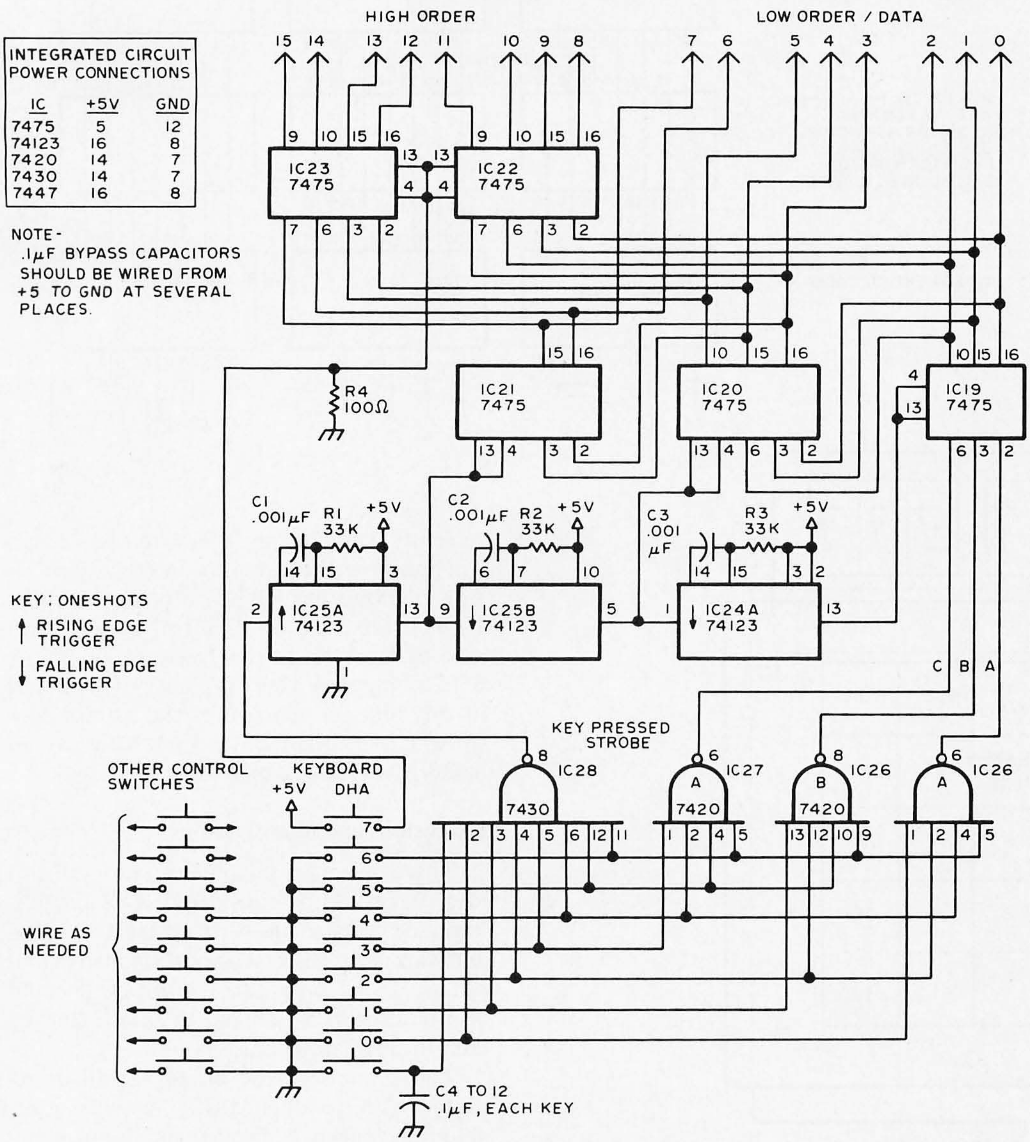


Figure 2: Keyboard Decode and Latching Circuits. The NAND gates are used to encode keystrokes which are entered into the low order at IC19. Shifting action controlled by oneshots is used to move data out of IC19 into IC20 then into IC21 (see text for description). A second 8 bit latch, manually controlled, is formed by ***IC22 and IC23*** *and is used to define high order address information.*

used to define bits 8 to 15 for use in addressing the memory of my computer.

The wiring of the displays is shown, along with a power regulator in figure 3. Each byte of information which is to be visible has a similar set of displays.

Construction

The keyboard was constructed on a 5 inch by 5 inch (12.7 cm by 12.7 cm) piece of printed circuit board material, with connections along one edge. A wire wrap board made out of Vector 0.1 inch grid (2.54 mm grid) stock and wire wrap sockets was used to mount integrated circuits; a small board was also fabricated to hold the display circuits. Figure 4 shows a top view and a side view of the mechanical arrangements in my original.

Bypass capacitors of 0.1 μF were used liberally between the power supply and ground. 0.1 μF capacitors were also mounted from the switch outputs to ground. These bypass capacitors were found necessary to keep from getting extra entries for single keystrokes. If the oneshots are inad-

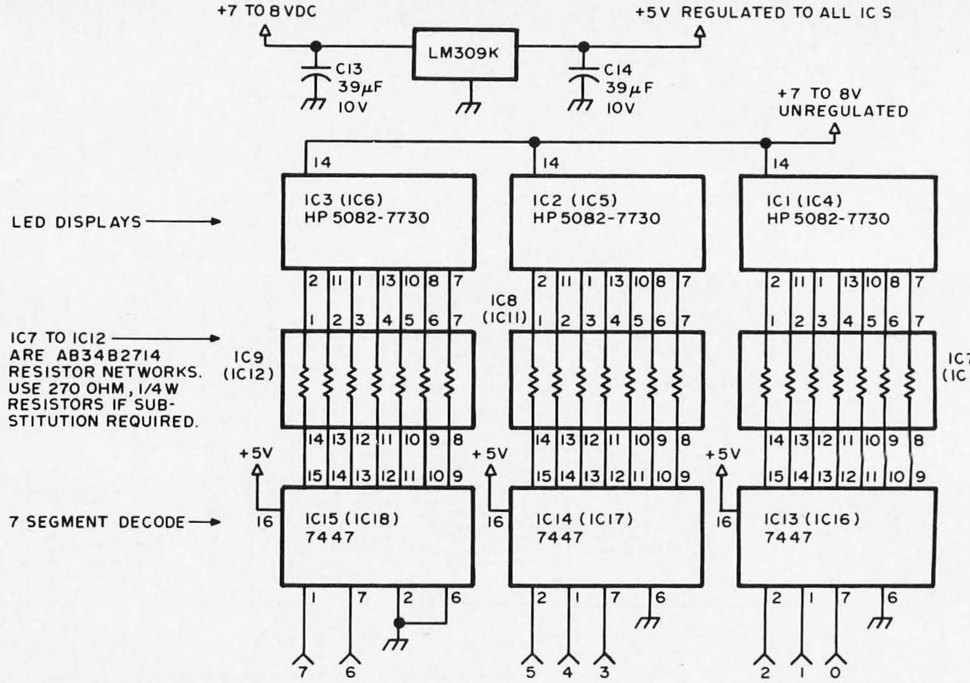


Figure 3: 3 Digit Octal Display and Power Regulator. One such display is required for each byte; the 9th bit is wired to zero for the purposes of decoding the high order digit.

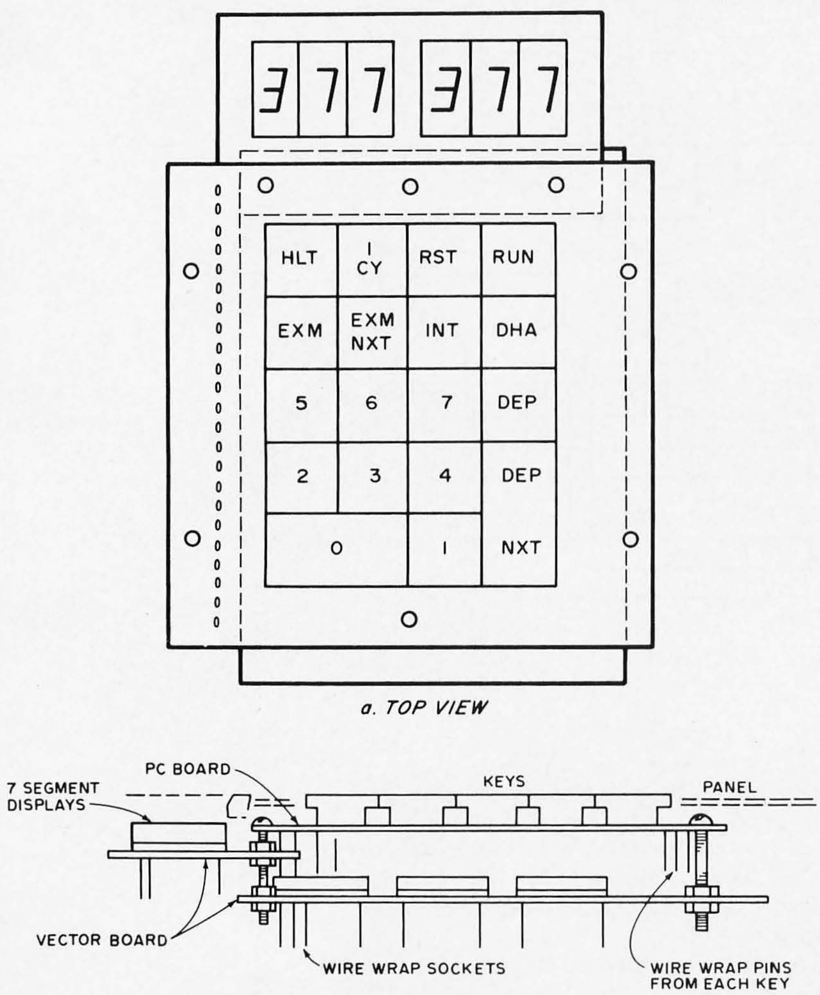


Figure 4: Mechanical Layout of the Author's Unit. The placement of parts is shown schematically in this illustration. The octal keys are at the lower left portion of the keyboard; the remaining keys are function buttons for the author's 6502 computer system.

vertently triggered on a key release, a zero will appear as the last entry in the display.

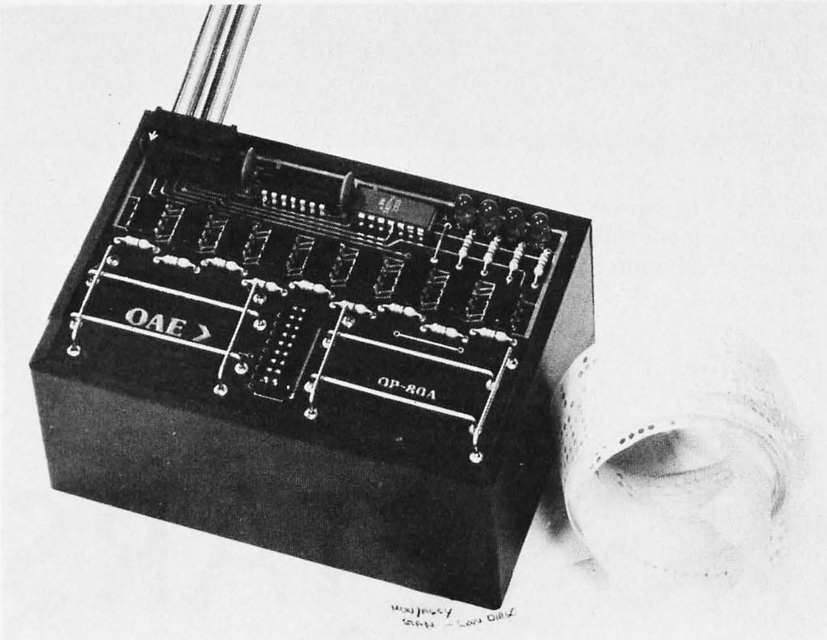
I mounted this keyboard on an 8 inch by 10 inch (20.3 cm by 25.4 cm) sloping panel box that serves as the front panel of my 6502 computer. There is plenty of room left in this box for the rest of the manual controls. I plan eventually to put address and data lights on this panel also.

Extensions and Modifications

There are many possible variations of this idea that could be considered. This design is intended to take the place of the 16 address and data switches that appear on many computers. Using only three displays, it could also be used as a terminal to enter 8 lines of data through an IO port.

The seven segment display could be replaced with 16 single LEDs, or with a suitable hexadecimal display, or even eliminated. If you have low order address switches which are separate from data switches, then a third 8 bit register and display just like the high address could be added to make a total of 24 lines.

I used a 7 to 8 volt unregulated power supply with a regulator for the ICs as shown in figure 3. The unregulated voltage was run directly to the display LEDs. If you use the computer's 5 volt supply for the displays as well as logic, then the 270 O limiting resistors can be replaced with 150 O resistors. The total current required for the circuit is about 1.5 amperes in either case. If you have a regulated supply, the regulator circuit of figure 3 can be omitted.■



What's New?

A New Low for Paper Tape Input Costs. . .

One way to lower the cost of mechanically oriented peripherals such as paper tape readers, printers, tape recording, etc, is to minimize the number of mechanical parts required. Here is an example of the result of such a strategy, the OP80A Low Cost Paper Tape Reader manufactured by Oliver Audio Engineering. There are "no moving parts" – except the paper tape itself when it is pulled through the guides as part of a read operation.

The device is designed to interface with an 8 bit parallel IO port for any of the microprocessors now on the market. It is ideal for loading all types of microprocessor support software in cases where it is distributed on paper tape. A well known 12K extended BASIC package is said to be loadable in 30 seconds (by experienced tape pullers). The electronic limits set a maximum feasible rate of 5000 CPS, with the actual rate entirely determined by the speed at which you pull the tape past the precision optical sensor array. If you make or buy a high speed paper tape winder, your rate of transfer could be quite high.

The device is available as a fully assembled and tested version for $95 or as a kit for $74.50. The device has four status LEDs, an anodized aluminum box, four feet of flat interface cable, assembly instructions (for the kit), interface schematics and related documentation. Oliver Audio Engineering is located at 1143 North Poinsettia Dr, Los Angeles CA 90046, (213) 874-6463.■

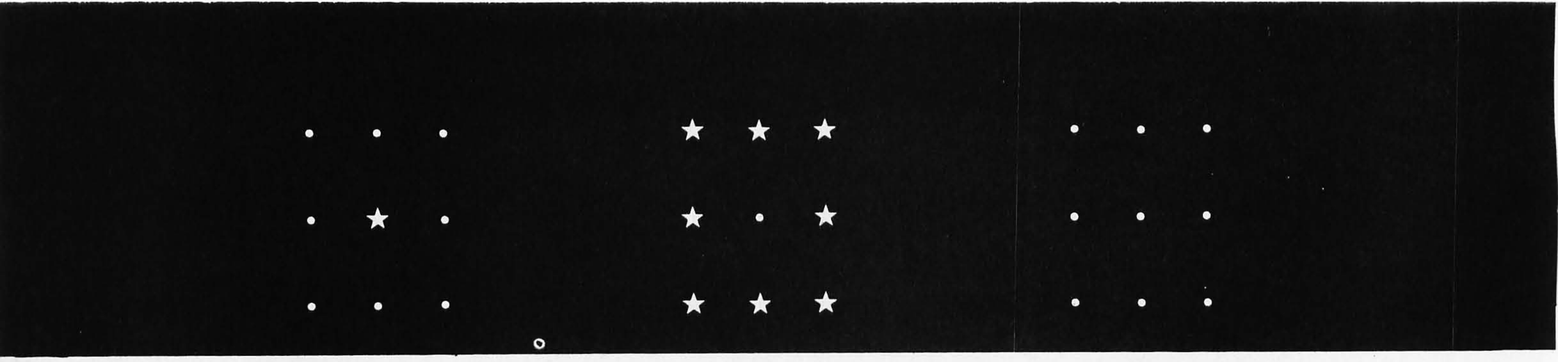


Figure 1: Three special patterns of stars and black holes. The game begins with a single star representing the Big Bang theory (left), and is won when the pattern of only one central black hole is achieved (center). The pattern shown on the right represents a loss and terminates the game.

SHOOTING STARS

Willard I Nico
Delta t
11020 Old Katy Rd, Suite 204
Houston TX 77043

There are probably as many reasons to have a computer in the home as there are computers in homes. For whatever reason you have one though, it's only human nature to want to show it off to other people.

Say you have a super program called "Investment Portfolio Analysis and Statistical Summary" (IPASS) up and running on your Scelbi 8H or whatever. It took months to write and debug the program and it involved several unique concepts of which you are justifiably proud. You can picture the furious activity going on inside the little heart of the computer and would dearly love to show off your skill to Mr and Mrs Nexdor and bask in their admiration. So you invite them over for cocktails.

The program runs flawlessly and, as the results flash on the display screen, you step back slightly to receive your praise. Mr Nexdor looks at you with a blank expression and says, "But will it grind pepper?"

That actually happened to me. One way around this problem is to save IPASS for your own enjoyment and have a game program or two available to show off. Of course, for some people game programs are the primary interest in having a home computer. Whatever your games interest, I think you'll find SHOOTING STARS an interesting addition to your library.

I started my quest for a "show-off" game about a year ago, searching everywhere for one that was just right. I learned a very interesting fact quickly: My computer doesn't speak BASIC, and to date many games have been written and published in that language.

So I had to do it myself. The result is SHOOTING STARS, a game with enough challenge to intrigue, enough variables to make learning to win difficult (but not impossible), and a couple of goodies thrown in to involve the player with the computer.

A complete program listing for 8008 computer is included, as well as the various messages that allow the computer to interact with the player.

The Game

Nine dot or asterisk characters are arranged in a 3 by 3 matrix on the playing field which may be shown on a CRT screen. The matrix represents the universe; asterisks are stars and dots are black holes. The player shoots stars which die and turn into black holes. When a star dies, it affects other stars and black holes in its particular galaxy.

How To Play

Each position in the universe is assigned a number (see figure 2). The computer outputs the current composition of the universe and asks YOUR SHOT? The player responds by typing the position number of the star he decides to shoot. Then the new constellation is displayed for the next shot.

Effect Of Shooting A Star

When a star dies, it affects the stars and black holes of its particular galaxy. The *effect* is that fragments of the star move into black holes to become new stars and other fragments collide with other stars and knock them out of orbit producing black holes. Each star has its own galaxy as shown in figure 3.

The Program

The game proceeds in an orderly manner which is shown in the Flow Chart of figure 4. The heading, rules and interactive messages require approximately 1600 B of memory. I use a Delta t Digital Recorder for message storage and retrieval since it operates in the reverse as well as forward incremental modes. Each message is prefaced with a message number surrounded with STX and ETX characters. A search routine in the main program finds the first address, decides whether the desired message is ahead or behind the current tape position, and rewinds or spins forward as necessary.

Table 1 is a list of the interactive messages. For computers with limited memory the essential messages are in the first portion of the table; the fancy heading is next, and the rules of the game occupy the largest number of bytes at the end of the text.

When the program is entered at address 014000, the 8008's H and L pointers are set to the beginning of the heading. Then the message control routine is called. It outputs sequentially each character of the message until the EM delimiter is encountered which returns control to the main program.

The status of the universe is stored in the B and C registers. Universe positions 1 through 4 and 6 through 9 are represented by the eight bits in the B register. A one bit represents a star, and a zero a black hole. Bit 0 of the C register keeps track of position 5.

The universe is set up in the beginning by clearing the B register and setting C to 001

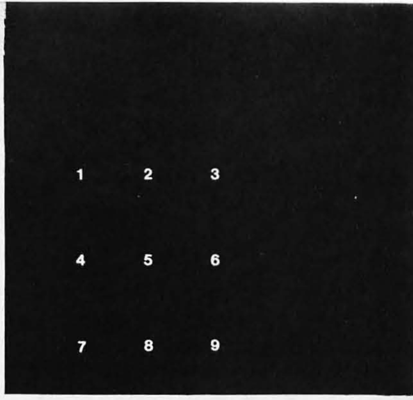
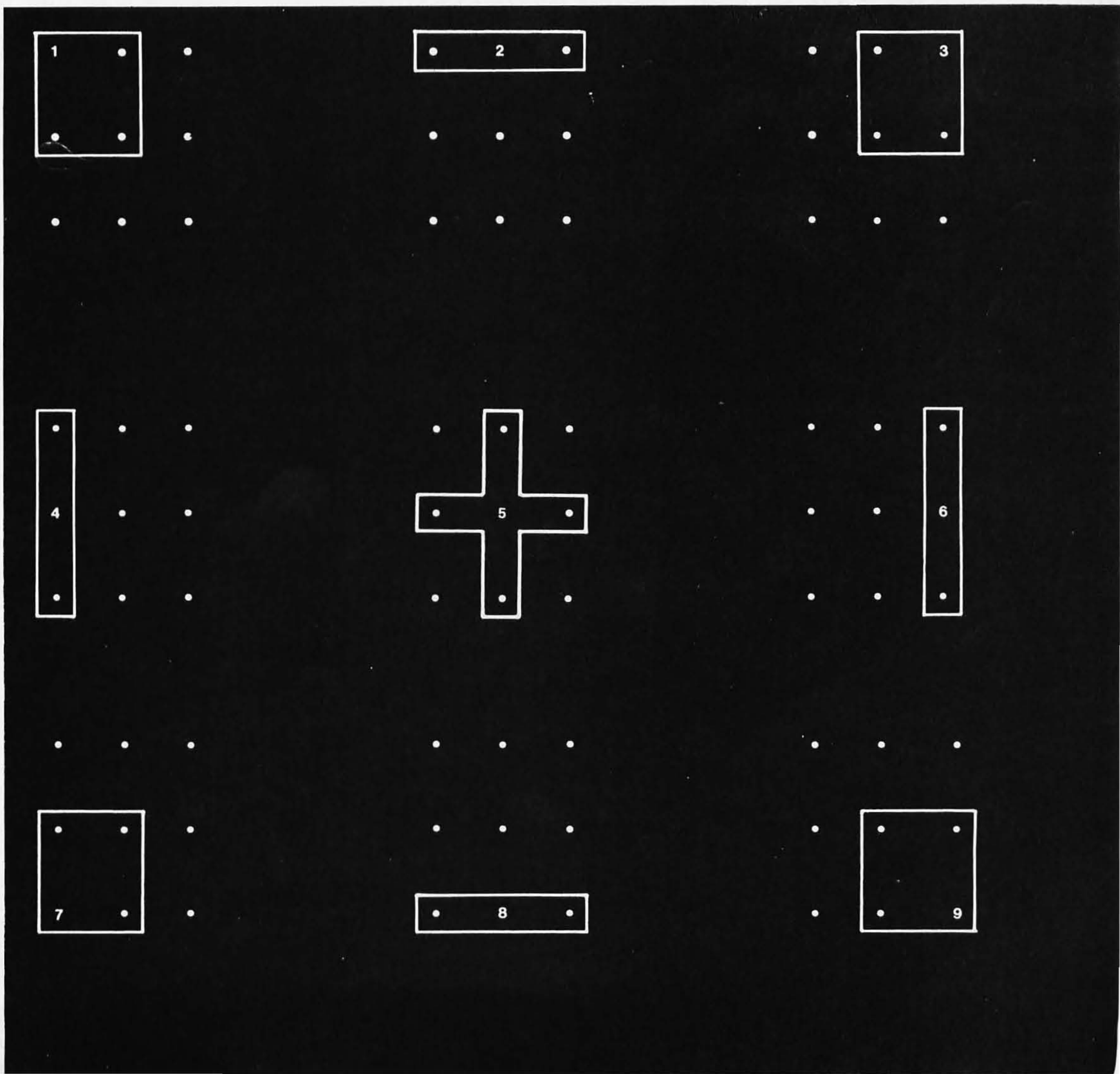


Figure 2: Positions in the universe are identified by numbers.

Figure 3: A complete set of galaxies which are associated with every star or black hole position. Stars or black holes within a galaxy are affected whenever the respective position has been chosen.



octal. The D register, which will tally the number of shots fired, is also cleared as part of the initialization process. Each time the print universe routine is entered after a valid shot, the D register is incremented to count the shot.

Displaying The Universe

First, the print universe routine is entered. This routine sets the E register to octal 012 and will decrement the register each time the print loop is executed. The E register tells the program when it needs to insert a couple of linefeeds for spacing, when it needs to branch to the position 5 special routine, and when it has finished printing the universe. These events occur at the following E register exception counts:

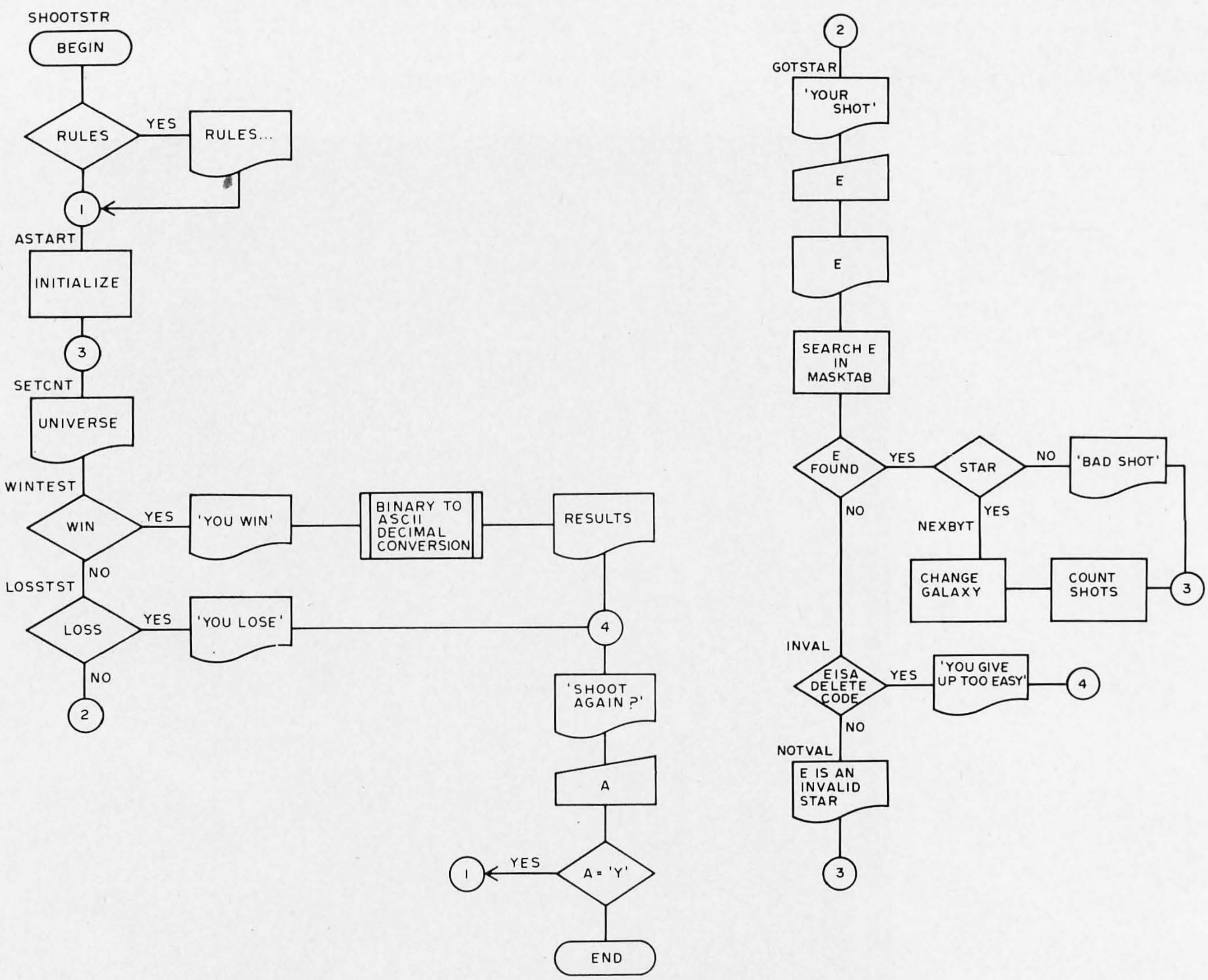
006 – Insert two linefeeds
005 – Go to position 5 subroutine
003 – Insert two linefeeds
000 – Done Print; exit

In normal processing, the positions represented by the bits in the B register are inspected one-by-one for star or black hole status, and the corresponding symbol is printed. It's done like this: The B register is loaded to A and rotated one place to the right. The rotated byte is loaded into B to be ready for the following position next time around in the loop. The carry flag is then tested for a one or zero. If the carry is zero, the program jumps to the dot output section. A one in the carry bit causes the asterisk output to be executed.

At the exception counts, further processing is required.

Thus when the E register count indicates that position 5 is the next one to be printed, the program loads the C register to A and

Figure 4: A flow chart of the SHOOTING STARS program acts as a guide to the listing. The labels indicated on this flow chart correspond to the labels found in table 3.



rotates the least significant bit to carry. The program then jumps back to the asterisk and dot output portion of the loop. Note that the rotated C register content is not loaded again to C, since we are only interested in the least significant bit.

Shoot A Star

When the universe has been displayed, the message YOUR SHOT? is printed and the computer waits for the player to type a number from 1 to 9 which indicates the star he wants to shoot. The ASCII code for the number the player types is compared to the first byte in each group of four contained in the MASKTAB table 2. The number of tries at the table is monitored by the E register, which starts at 011 and is decremented each time around the "test for match" loop. If the E register gets to 000 without finding a match, the input is tested for code 177 (delete), indicating that the player gives up and wants to start over. If a match still can't be found, the NOT A VALID STAR NUMBER message is printed, and the universe displayed again. If this happens, the print universe routine is entered just after the instruction that causes the shot to be counted, so the player won't be charged for his mistake.

When a find is made in the MASKTAB table, the program is ready to process the player's shot. First, it must make sure the player is following the rules and hasn't shot a black hole. The second byte of the four byte group is used as a "mask" to blank out all the positions of the universe except the one that has been shot. Figure 5 shows how the mask is used with the Boolean AND function to isolate the bit representing the shot position from among the eight bits of the B register. After masking out all but the selected position, the resultant byte is tested to see if it is zero. If it is, the shot position was a black hole and the message HEY! YOU CAN ONLY SHOOT STARS, NOT BLACK HOLES! is printed. If this happens, the universe is displayed again without counting the shot.

If the mask itself is zero, it indicates that position 5 was selected, and so the program

Table 1: Program Messages. This table lists all the messages used by SHOOTING STARS. Each message entry in the table starts with a symbolic name and an absolute address. The text should be stored at ascending memory address locations, and terminated with an end of message (EM) delimiter of octal 031, which is printed as ■. The symbolic names in this table are referenced by table 3.

```
MESS1:
      016000
      HEY! YOU CAN ONLY SHOOT STARS,
      NOT BLACK HOLES.
      TRY AGAIN!
      ■

MESS2:
      016077
      THAT WASN'T A VALID STAR NUMBER!
      TRY AGAIN!
      ■

MESS3:
      016156
      YOU LOST THE GAME!
      WANT TO SHOOT SOME MORE STARS?   ■

MESS4:
      016243
      YOU WIN! GOOD SHOOTING!
      YOU FIRED   ■

MESS5:
      016310
       SHOTS.
      BEST POSSIBLE SCORE IS 11 SHOTS.
      WANT TO SHOOT AGAIN, DEADEYE?   ■

MESS6:
      017022
      YOU GIVE UP TOO EASILY!
      WANT TO SHOOT SOME MORE STARS?.  ■

MESS7:
      017114
      YOUR SHOT?   ■

HMESS:
      017131
      S  H  O    SSS     TTT     AAA     RRR     SSS
                 S        T      A A     R R     S
      O  •  T    SSS      T      AAA     RRR     SSS
                   S      T      A A     RR        S
      I  N  G    SSS      T      A A     R R     SSS
      ••••••••••••S H O O T I N G   S T A R S••••••••••••
                    A BRAIN TEASER GAME
      WANT THE RULES?   ■

PAGE1:
      020147
      THERE ARE STARS:      •
      AND BLACK HOLES:      .
      IN THE UNIVERSE:         •  •  •

                               •  •  •

                               •  •  •

      YOU SHOOT A STAR
      (NOT A BLACK HOLE)
      BY TYPING ITS NUMBER        1 2 3
                                  4 5 6
                                  7 8 9
      THAT CHANGES THE STAR TO A BLACK HOLE!
      (TO SEE MORE RULES, TYPE ANY KEY.) ■

PAGE2:
      021277
      EACH STAR IS IN A GALAXY. WHEN YOU
      SHOOT A STAR, EVERYTHING IN ITS GALAXY
      CHANGES. ALL STARS BECOME BLACK HOLES
      AND ALL BLACK HOLES BECOME STARS.
      GALAXIES:
      1 • .    • 2 •    . • 3    • . .    . • .
      • • .    . . .    . • •    4 . .    • 5 •
      . . .    . . .    . . .    • . .    . • .

      . . •    . . .    . . .    . . .
      . . 6    • • .    . . .    . • •
      . . •    7 • .    • 8 •    . • 9
      (TYPE ANY KEY FOR LAST PAGE OF RULES.) ■

PAGE3:
      023137
      THE GAME STARTS          . . .
      WITH THE UNIVERSE        . • .
      LIKE THIS
                               . . .

      YOU WIN WHEN YOU         • • •
      CHANGE IT TO THIS        • . •

                               • • •

      YOU LOSE IF YOU          . . .
      GET THIS
                               . . .

                               . . .
      READY TO PLAY. TYPE ANY KEY TO START
      THE GAME. GOOD LUCK! ■
```

tests the C instead of the B register for a star.

Change A Galaxy

Once the program has determined that the shot was valid, it can use the next byte in the MASKTAB table to change the dots and stars in the galaxy of the "shot" star. Again, the table entry is a mask, but this time the Boolean EXCLUSIVE OR function is used. The result is that the selected positions are *complemented;* one bits are changed to zero bits and the zeros are changed to ones. Figure 6 shows how the mask does this neat trick. After the change is made, the new universe is stored in the B register.

Byte four of the MASKTAB table entry contains a mask that is used to EXCLUSIVE OR the C register to change position 5 if required. If star 5 is to be complemented, the mask will be octal 001; if not, it will be octal 000.

After the universe in the B and C registers is changed, the new universe is displayed and the cycle repeats until a win or a loss is detected, or until the player gives up.

Win Or Loss Test

Each time the universe is displayed, it is tested for a win or a loss. If both the B and C registers contain the octal number 000, the YOU LOST THE GAME message is printed, and the opportunity to play again is offered.

If the B register contains octal 377 and C is octal 000 a win is detected. After displaying the proper message, the binary content of the D register is converted to decimal numbers and the number of shots fired is printed. The calculation is performed by the binary to decimal conversion subroutine.

Binary To Decimal Conversion

The B, C and E registers are assigned the functions of summing the hundred, ten and unit digits of the score respectively. The process is one of repetitively adding a one to the three digit number while subtracting a one from the shots fired register (D). Looping continues until all shots fired have been counted in the 3 digit decimal form.

The somewhat unusual feature of the binary to decimal conversion is that it is done directly in ASCII numeric code. The three registers B, C and E are intially loaded with octal 060, which is the ASCII numeric character zero. After each increment, the least significant digit register (E) is tested to see if it contains octal 072. If it does, the register has counted 060, 061 ... 071, which is 0 through 9 in ASCII, and has just been incremented one more to 072. When

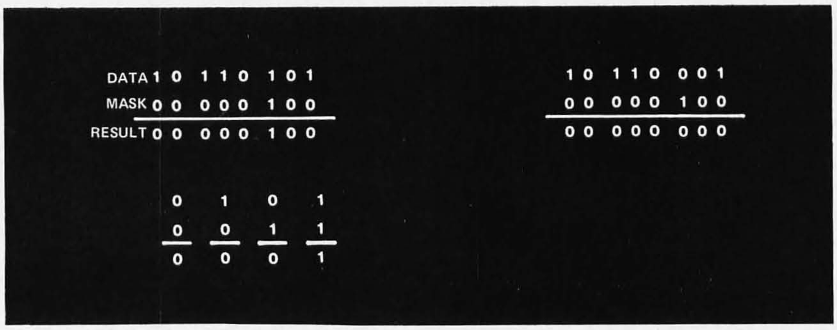


Figure 5: The AND function of Boolean logic is used to mask the current universe in order to select one position for testing each shot.

	LOCATION	SHOT	POSITION MASK	GALAXY MASK	CENTER MASK
MASKTAB	015070	061	001	013	001
	015074	062	002	007	000
	015100	063	004	026	001
	015104	064	010	051	000
	015110	065	000	132	001
	015114	066	020	224	000
	015120	067	040	150	001
	015124	070	100	340	000
	015130	071	200	320	001

Table 2: MASKTAB, a table of masks to test and alter galaxies. This table gives the data needed for memory locations 015/070 to 015/133 in the SHOOTING STARS program. This table is used to check the shot fired for a valid star number and to change the portion of the universe which is affected by the star's change.

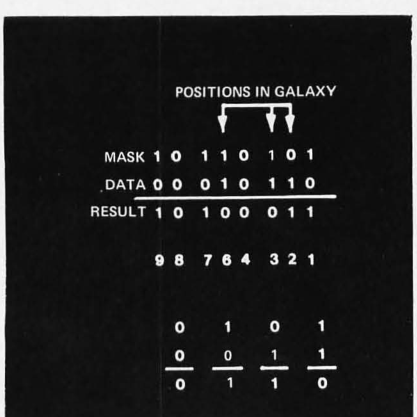


Figure 6: The EXCLUSIVE OR function of Boolean logic is used to complement bits selected according to the galaxy information stored for the position just shot.

the register has 072, a carry condition exists. When this condition is detected, the register is reset to 060 and the next register in line (C) is incremented. After incrementing, the second register is tested for a carry in the same manner, and so on. When all the shots have been counted, the registers B, C and E will not only represent the decimal equivalent of the shots fired, but will contain the proper ASCII codes for the decimal digits of the count.

Print The Shots

To suppress leading zeros, the hundreds digit (B) is tested for octal 060. If it contains any other code, the contents of all three registers will be printed. If it contains octal 060, the tens register (C) is similarly tested and the output will be one digit if it is at zero (code 060) and two digits if it is not.

Figure 7 contains a flow chart of the binary to decimal conversion program. You may find use for it in some of your other programs.

Program Listing Conventions

Table 3 contains the complete program as it was implemented in my 8008 system using the SCELBI 8H computer. The listing is in symbolic assembly language with absolute octal address and memory contents.

The 8008 computer has 8 possible restart instructions which are one byte calls to locations in the first portion of memory address space. These are used to access utility subroutines needed by the SHOOTING STARS program. The required restarts are as follows:

RST0: User's input routine, starting at location 000/000 which is used to wait for one character input from the keyboard device.

RST1: Exit Routine, starting at location 000/010. This is a return address to the system monitor for the computer.

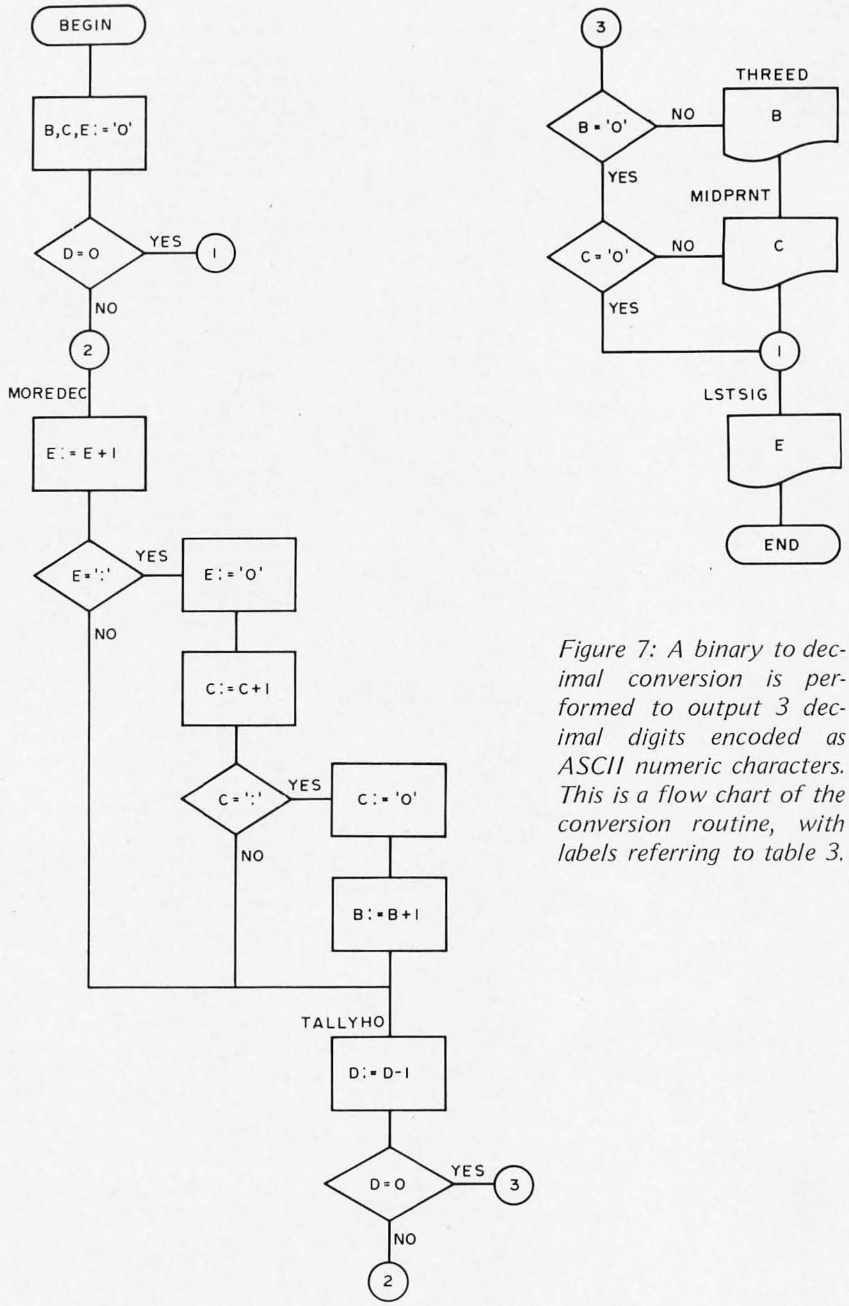


Figure 7: A binary to decimal conversion is performed to output 3 decimal digits encoded as ASCII numeric characters. This is a flow chart of the conversion routine, with labels referring to table 3.

octal address	octal code	label	op.	operand	commentary
014/000	006 012	SHOOTSTR	LAI	012	display a linefeed to
014/002	025		RST	2	initialize display;
014/003	066 131		LLI	L(HMESS)	set address pointers
014/005	056 017		LHI	H(HMESS)	to heading message;
014/007	106 134 015		CAL	OUTPUT	print message & return;
014/012	106 151 015		CAL	INPUT	call input looper;
014/015	074 116		CPI	'N'	is first letter 'N'?
014/017	150 052 014		JTZ	ASTART	if so then plunge into game;
014/022	066 147		LLI	L(PAGE1)	if not then point to first
014/024	056 020		LHI	H(PAGE1)	page of rules text;
014/026	106 134 015		CAL	OUTPUT	and go output rules message;
014/031	075		RST	7	wait for goahead;
014/032	066 277		LLI	L(PAGE2)	point to second page of
014/034	056 021		LHI	H(PAGE2)	rules text;
014/036	106 134 015		CAL	OUTPUT	display second page of rules;
014/041	075		RST	7	wait for goahead;
014/042	066 137		LLI	L(PAGE3)	point to third page of
014/044	056 023		LHI	H(PAGE3)	rules text;
014/046	106 134 015		CAL	OUTPUT	display third page of rules;
014/051	075		RST	7	wait for goahead;
014/052	006 012	ASTART	LAI	012	set up linefeed;
014/054	025		RST	2	display one linefeed,
014/055	025		RST	2	then a second linefeed,
014/056	025		RST	2	then a third;
014/057	016 000		LBI	0	initialize the universe
014/061	026 001		LCI	1	to starting pattern;

Table 3: The SHOOTING STARS program specified in symbolic assembly language with an absolute listing of addresses and codes for the author's system.

octal address	octal code	label	op.	operand	commentary
014/063	331		LDB		then clear shot counter;
014/064	030	CNTSHOT	IND		count a shot (anticipatory);
014/065	046 012	SETCNT	LEI	10D	loop count 10 iterations;
014/067	041	DISLOOP	DCE		is the loop done?
014/070	150 321 014		JTZ	WINTEST	if so then go to win testing;
014/073	304		LAE		if not then continue display;
014/074	074 006		CPI	6	is it fourth cycle?
014/076	150 142 014		JTZ	LINFEED	if so then new line needed;
014/101	074 003		CPI	3	is it seventh cycle?
014/103	150 142 014		JTZ	LINFEED	if so then new line needed;
014/106	074 005		CPI	5	is it star number 5?
014/110	150 151 014		JTZ	FIVTST	if so then go test star 5;
014/113	250	NEDOT	XRA		clear the carry (and A too);
014/114	301		LAB		move universe to A;
014/115	012		RRC		rotate next place into carry;
014/116	310		LBA		save it in B for a while;
014/117	100 130 014	PSEUDOT	JFC	LOADOT	if dot then go output dot;
014/122	006 052		LAI	'*'	otherwise load a star;
014/124	025		RST	2	then print the star;
014/125	104 133 014		JMP	SPCNOW	branch around dot logic;
014/130	006 056	LOADOT	LAI	'.'	load a dot;
014/132	025		RST	2	then print the dot;
014/133	006 040	SPCNOW	LAI	' '	load a space;
014/135	025		RST	2	print one space,
014/136	025		RST	2	then print a second;
014/137	104 067 014		JMP	DISLOOP	waltz around loop once more;
014/142	006 012	LINFEED	LAI	012	load a line feed;
014/144	025		RST	2	display a line feed,
014/145	025		RST	2	then a second one;
014/146	104 113 014		JMP	NEDOT	back to print next dot or star;
014/151	250	FIVTST	XRA		no operation intended – leftover;
014/152	302		LAC		get position 5 status;
014/153	012		RRC		put status into carry;
014/154	104 117 014		JMP	PSEUDOT	rejoin main line after RRC;
014/157	006 012	GOTSTAR	LAI	012	load a line feed;
014/161	025		RST	2	have finished universe print,
014/162	025		RST	2	so print several
014/163	025		RST	2	line feeds
014/164	025		RST	2	to separate
014/165	025		RST	2	successive rounds;
014/166	066 114		LLI	L(MESS7)	point to the 'your shot'
014/170	056 017		LHI	H(MESS7)	message;
014/172	106 134 015		CAL	OUTPUT	then go print it;
014/175	005		RST	0	call input for character;
014/176	025		RST	2	immediately echo the input;
014/177	340		LEA		save input temporarily in E;
014/200	006 012		LAI	012	load a line feed;
014/202	025		RST	2	print three line feeds to
014/203	025		RST	2	space out the response
014/204	025		RST	2	a bit more;
014/205	304		LAE		recover input for testing;
014/206	046 011		LEI	9D	loop count for table search;
014/210	066 070		LLI	L(MASKTAB)	set up pointer to the
014/212	056 015		LHI	H(MASKTAB)	the mask table;
014/214	277	NEXGRUP	CPM		is input equal table character?
014/215	150 233 014		JTZ	FOUND	if so then go alter structure of
014/220	041		DCE		the universe otherwise just
014/221	150 273 014		JTZ	INVAL	check end of loop;
014/224	060		INL		increment the L
014/225	060		INL		register pointer
014/226	060		INL		four times to get
014/227	060		INL		to next table entry;
014/230	104 214 014		JMP	NEXGRUP	then go test next entry;
014/233	060	FOUND	INL		point to position mask
014/234	307		LAM		and load mask into A;
014/235	074 000		CPI	0	is it zero?
014/237	110 253 014		JFZ	UNIV2A	if not then fringe position;
014/242	302		LAC		otherwise the center position;
014/243	074 001		CPI	1	is a star in center?
014/245	110 165 015		JFZ	BADFELO	if not then have wrong move;
014/250	104 260 014		JMP	NEXBYT	if so then go process star;
014/253	301	UNIV2A	LAB		rest of universe to A;
014/254	247		NDM		AND with mask to isolate star;
014/255	150 165 015		JTZ	BADFELO	if not star then wrong move;
014/260	060	NEXBYT	INL		point to the galaxy mask;
014/261	301		LAB		fetch universe again;
014/262	257		XRM		and complement the universe
014/263	310		LBA		on a fine performance;
014/264	060		INL		point to center mask;
014/265	302		LAC		fetch center of universe;
014/266	257		XRM		complement center if required;
014/267	320		LCA		save center of universe;
014/270	104 064 014		JMP	CNTSHOT	go display a new universe;
014/273	074 177	INVAL	CPI	177	was invalid shot a 'delete'?
014/275	110 307 014		JFZ	NOTVAL	if not then recycle bad star;
014/300	066 022		LLI	L(MESS6)	otherwise point to giving up
014/302	056 017		LHI	H(MESS6)	message;
014/304	104 034 015		JMP	PRNTIT	display then test for restart;
014/307	066 077	NOTVAL	LLI	L(MESS2)	point to the invalid star
014/311	056 016		LHI	L(MESS2)	number message
014/313	106 134 015	OUTMES	CAL	OUTPUT	output a message then
014/316	104 065 014		JMP	SETCNT	go display the universe again;
014/321	301	WINTEST	LAB		move universe to A;
014/322	074 377		CPI	11111111B	are all fringe stars present?
014/324	110 050 015		JFZ	LOSSTST	if not see if player has lost;
014/327	302		LAC		fetch center of universe;
014/330	074 000		CPI	0	is center of universe empty?
014/332	110 157 014		JFZ	GOTSTAR	is full then not win;
014/335	066 243		LLI	L(MESS4)	no star! got a win, folks
014/337	056 016		LHI	H(MESS4)	so point to win message;
014/341	106 134 015		CAL	OUTPUT	then display win message;
014/344	046 060		LEI	'0'	begin binary to decimal conversion
014/346	314		LBE		by setting all three working
014/347	324		LCE		register to (ASCII) zero;
014/350	031		DCD		get rid of last shot;
014/351	303		LAD		move shot count to A for test;
014/352	074 000		CPI	0	test for zero (not needed in
014/354	150 026 015		JTZ	LSTSIG	SHOOTING STARS but generally
					useful with conversions);
014/357	006 072		LAI	':'	need compare to ASCII '9' + 1;
014/361	040	MOREDEC	INE		count up one in 1.s. digit;
014/362	274		CPE		is it equal to overflow code?
014/363	110 000 015		JFZ	TALLYHO	if not then tally and continue;
014/366	046 060		LEI	'0'	else reset 1's digit to zero
014/370	020		INC		and carry into next digit;

RST2: User's output routine, starting at location 000/020. This routine prints or displays one character on the output device for the system. The character to be output is in the A register when RST2 is entered.

RST7: A "do Nothing" keyboard input acknowledgement routine, starting at location 000/070. Any character typed on the keyboard causes return from this subroutine.

For the optimum use of the program, the output device should be a cathode ray tube terminal with a scrolling feature.

Game Background

I first saw the SHOOTING STARS game in the September, 1974, issue of PCC† as a program called TEASER. If you are an analytical person, you can figure out all of the possible positions.

PCC Editor, Bob Albrecht, told me that the program was contributed to the Hewlett-Packard software library, and originally written in BASIC.■

†PCC is People's Computer Company which publishes a tabloid size computer hobbyist newspaper five or more times during the school year. It's filled with games written in BASIC, art, and computer news. If you are interested, write to People's Computer Company, PO Box 310, Menlo Park CA 94025.

Symbol table, in order of appearance

SHOOTSTR	014/000
ASTART	014/052
CNTSHOT	014/064
SETCNT	014/065
DISLOOP	014/066
NEDOT	014/113
PSEUDOT	014/117
LOADOT	014/130
SPCNOW	014/133
LINFEED	014/142
FIVTST	014/151
GOTSTAR	014/157
NEXGRUP	014/214
FOUND	014/233
UNIV2A	014/253
NEXTBYT	014/260
INVAL	014/273
NOTVAL	014/307
OUTMES	014/313
WINTEST	014/321
MOREDEC	014/361
TALLYHO	015/000
THREED	015/023
MIDPRNT	015/025
LSTSIG	015/026
RECYC	015/032
PRNTIT	015/034
LOSSTST	015/050
MASKTAB	015/070
OUTPUT	015/134
INPUT	015/151
GETNEXT	015/154
BADFELO	015/165
MESS1	016/000
MESS2	016/077
MESS3	016/156
MESS4	016/243
MESS5	016/310
MESS6	017/022
MESS7	017/144
HMESS	017/131
PAGE1	020/147
PAGE2	021/277
PAGE3	023/137

octal address	octal code	label	op.	operand	commentary
014/371	272		CPC		is it equal to overflow code too?
014/372	110 000 015		JFZ	TALLYHO	if not then tally and continue;
014/375	026 060		LCI	'0'	else reset middle digit to zero
014/377	010		INB		and carry into m.s. digit;
015/000	031	TALLYHO	DCD		decrement score counter for tally;
015/001	110 361 014		JFZ	MOREDEC	if not zero then keep loopin;
015/004	301		LAB		fetch leading digit to A;
015/005	074 060		CPI	'0'	is it (ASCII) zero?
015/007	110 023 015		JFZ	THREED	if not go display three digits;
015/012	302		LAC		fetch middle digit to A;
015/013	074 060		CPI	'0'	is it (ASCII) zero too?
015/015	110 025 015		JFZ	MIDPRNT	if not go display two digits;
015/020	104 026 015		JMP	LSTSIG	if so display only one;
015/023	025	THREED	RST	2	display three digits, left first;
015/024	302		LAC		fetch middle digit to A;
015/025	025	MIDPRNT	RST	2	display two digits, left first;
015/026	304	LSTSIG	LAE		fetch 1's digit;
015/027	025		RST	2	display remaining digit;
015/030	066 310		LLI	L(MESS5)	point to first part of you win;
015/032	056 016	RECYC	LHI	H(MESS5)	second part of MESS5/MESS6 pointer;
015/034	106 134 015	PRNTIT	CAL	OUTPUT	display the message;
015/037	106 151 015		CAL	INPUT	fetch a character for continue
015/042	074 131		CPI	'Y'	query, is it "yes"?
015/044	150 052 014		JTZ	ASTART	if so then continue game;
015/047	015		RST	1	otherwise call EXIT;
015/050	074 000	LOSSTST	CPI	0	is fringe universe all black holes?
015/052	110 157 014		JFZ	GOTSTAR	if not then continue game;
015/055	302		LAC		if so then test center position;
015/056	074 000		CPI	0	is center also black hole?
015/060	110 157 014		JFZ	GOTSTAR	if not then continue game;
015/063	066 156		LLI	L(MESS3)	else point to loss message;
015/065	104 032 015		JMP	RECYC	and go print loss;
015/070	see Table II	MASKTAB	BLK	036D	36 bytes of mask table;
015/134	307	OUTPUT	LAM		fetch next message byte;
015/135	074 031		CPI	031	is it a delimiter?
015/137	053		RTZ		return when delimiter found;
015/140	025		RST	2	otherwise display byte;
015/141	060		INL		point to next byte;
015/142	110 134 015		JFZ	OUTPUT	is it page boundary?
015/145	050		INH		if so increment page;
015/146	104 134 015		JMP	OUTPUT	and then recycle;
015/151	005	INPUT	RST	0	get next character;
015/152	340		LEA		save it in E;
015/153	025		RST	2	echo on display;
015/154	005	GETNEXT	RST	0	get next character;
015/155	025		RST	2	echo on display;
015/156	074 012		CPI	012	was it a line feed?
015/160	110 154 015		JFZ	GETNEXT	if not continue scan;
015/163	304		LAE		if so, restore first input;
015/164	007		RET		and then return to caller;
015/165	066 000	BADFELO	LLI	L(MESS1)	point to the error message
015/167	056 016		LHI	H(MESS1)	admonishing bad 'star';
015 171	104 313 014		JMP	OUTMES	and go display error;

Notation:

L(HMESS) = low order 8 bits of address of HMESS;

H(HMESS) = high order 8 bits of address of HMESS;

'N' = the ASCII character "N";

9D = the decimal number 9;

7 = the octal number 7 (with high order zeros as needed);

mnemonics are from original Intel 8008 documentation;

octal code is shown in ascending address order top to bottom, left to right;

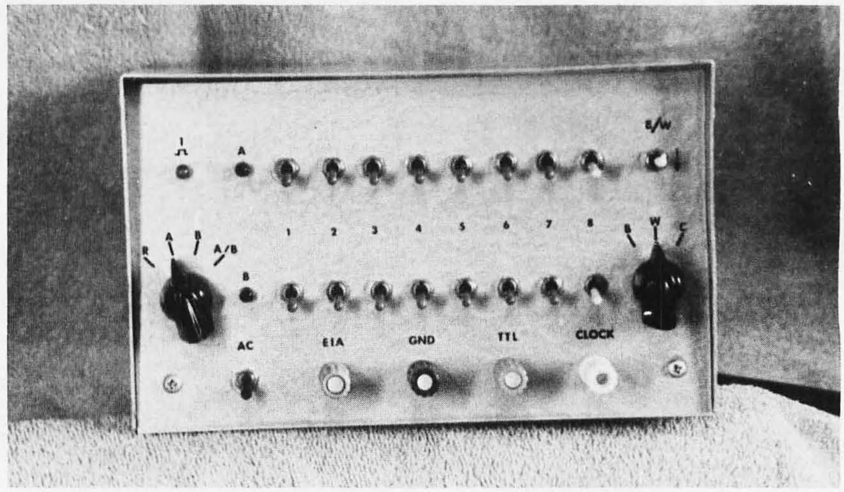


Photo 1: Front Panel View. The controls of the ASCII word generator include rotary mode selections at the left and right, two rows of data definition switches for words A and B, and various indicators.

Build a Serial ASCII Word Generator

Ronald J Finger
3417 E 65th Av
Anchorage AK 99502

A couple of months ago, I decided to design and build a hardware ASCII to Baudot code converter. The design of this unit was going along quite well when it suddenly dawned on me that there would be no way to test the converter without generating a serial ASCII test pattern. So the converter was put aside temporarily while I decided what to do about generating test patterns.

My first impulse was to throw together a quick breadboard setup that would just generate a serial ASCII output in the UART format. It turned out, however, that it would require a fair amount of logic just to do that. Why not add some versatility and make it a real test instrument? By now, the wheels had started to turn and some desired features were suggested by some of my associates. The final result is shown in figure 1 and photos 1, 2 and 3.

For the purpose of this article, I'm defining an ASCII word to be the complete sequence of 1 start bit, 7 data bits with an ASCII character code, 1 data bit with the ASCII parity information, and 2 stop bits. This is the standard asynchronous data format used for serial communications.

Output Patterns and Modes

The generator has four different data output patterns available. They are selected by the four position rotary switch (S19) on the left side of the panel as shown in photo 1. The selections are as follows:

- R – The only output available in this position is the "rubout." All 8 bits will be logic high regardless of any other switch setting.
- A – The pattern of this word is determined by the top row of bit selector switches (S1 to S8).
- B – This pattern is set by the bottom row of switches (S9 to S16).
- A/B – In this position, words A and B are selected alternately.

The generator has three different output modes. They are selected by the three position rotary switch (S20) on the right side of the panel. All of the modes produce a serial ASCII output. The difference is in the timing of the output:

- B – In this mode, the word is generated a single bit at a time. There is no limitation on the interval between bits. This mode is useful in the design and analysis of computer input output hardware.
- W – A single word will be produced in this mode with a rate of 110 baud. This mode proved to be very useful

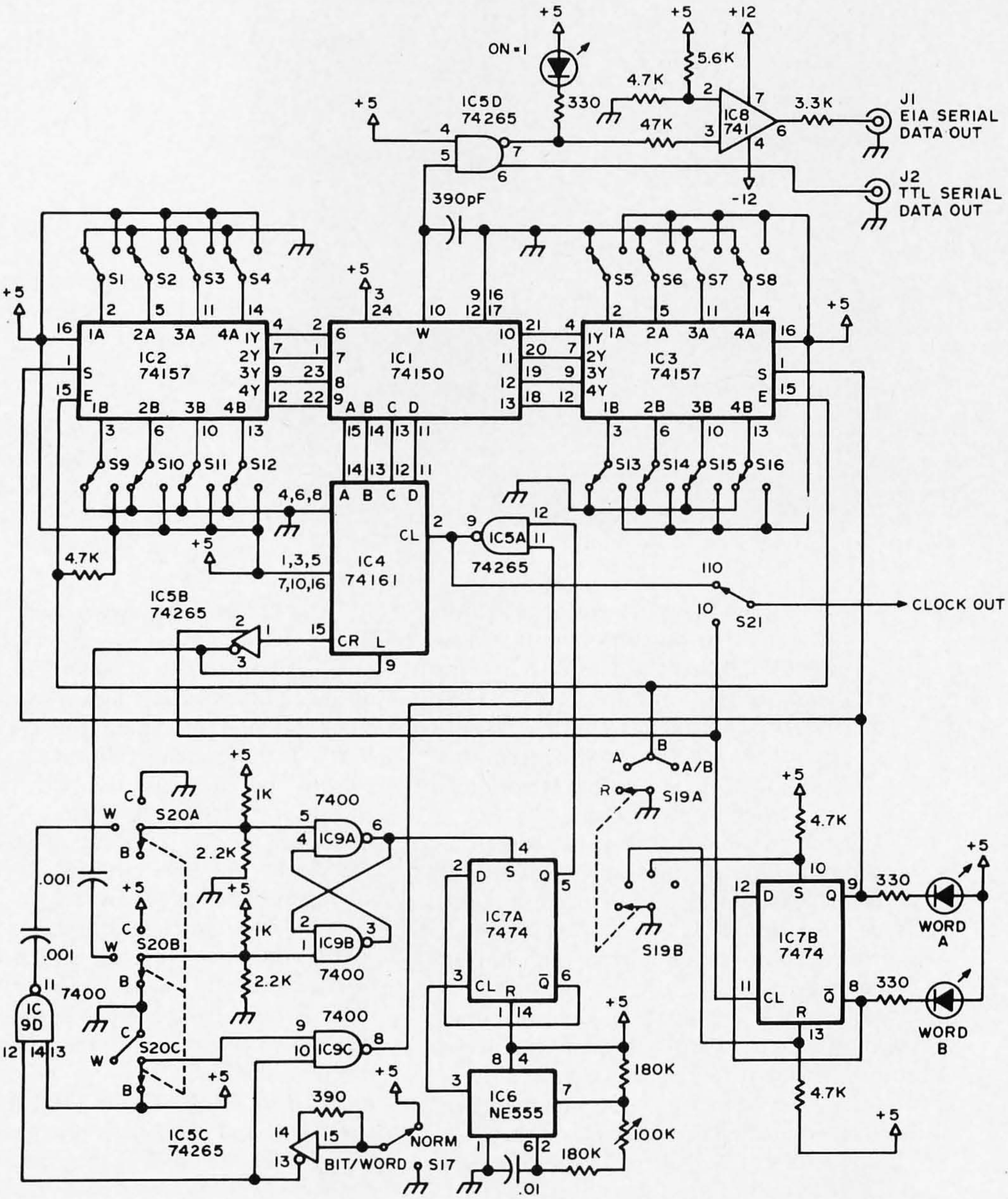


Figure 1: Circuit of the ASCII Word Generator. This design generates two different data words set by switches and features four patterns available in three operating modes. Power and ground connections for the integrated circuits are as follows:

Number	Device	+5 V	Ground
IC1	74150	24	12
IC2	74157	16	8
IC3	74157	16	8
IC4	74161	16	8
IC5	74265	16	8
IC6	555	8	1
IC7	7474	14	7
IC9	7400	14	7

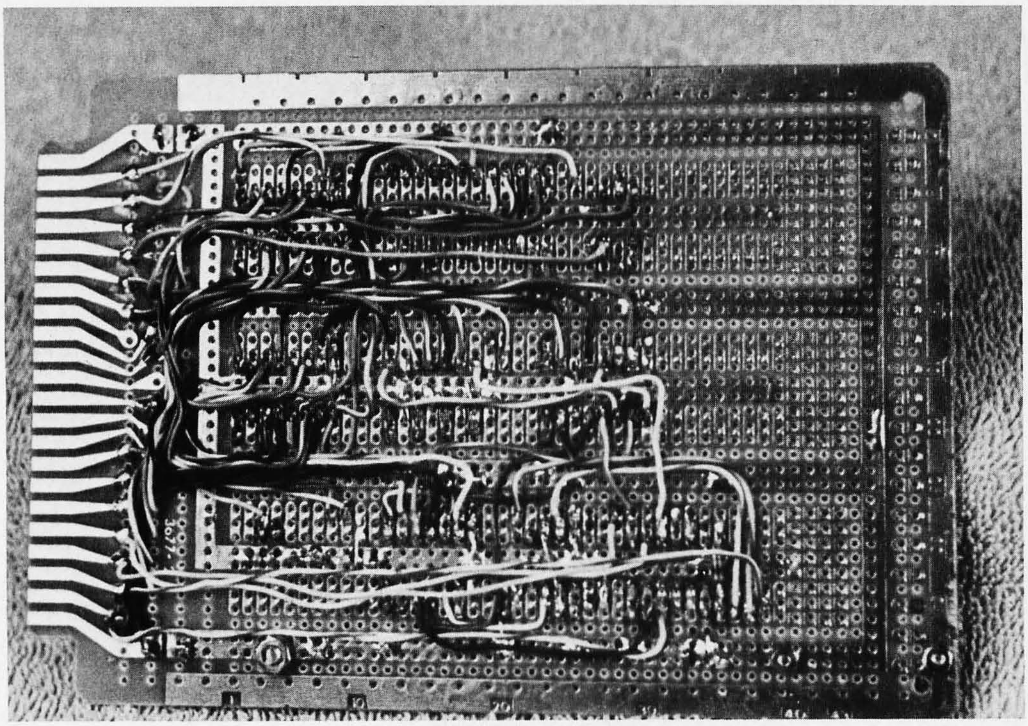


Photo 2: Wiring of the Circuit Board. The prototype word generator was built using a Vector 3677-2 DIP plugboard. Point to point wiring was done using solder for interconnection with Molex pins for the IC leads.

in testing the aformentioned ASCII to Baudot code converter. It is also useful for loading data into a computer in serial form.

C – This mode produces continuous output of the selected word pattern. It is most useful in testing teleprinters and other output devices.

The switch labeled B/W (S17) is used to trigger the single bit or single word output modes. It has no function in the continuous mode. A spring loaded toggle switch is shown in photo 1, although a push button could be used.

LED lamps are used to indicate either word A or B, and to display the serial output. In the bit mode, the output pattern can easily be verified by observing the LED, which is connected to the serial output.

Standard TTL and RS-232C outputs are available from the front panel binding posts. A choice of two different clock outputs is available. A symmetrical 110 baud and pulsed 10 baud output can be selected by S21. This switch is not used very often, so it is mounted on the rear panel. The clock outputs are very useful for the synchronization of an oscilloscope. The 110 baud will sync the bit rate and the 10 baud will sync the word rate.

The Circuit

Refer to the schematic diagram of figure 1 for a review of the circuit operation. IC6 is connected as an astable multivibrator, running at 220 Hz. The output is applied to IC7A, a D flip flop connected as a toggle to divide the frequency by two. This not only provides a perfectly symmetrical 110 baud output, but provides a means of turning the clock on and off using the set input of IC7A. The SR latch, IC9A and B, is used to control the clock via IC7A's S input. S20 and IC9D control the latch. In the continuous mode the latch output stays high so the clock runs uninterrupted. In the word mode, the latch is set by a pulse generated by S17. This switch is debounced by IC5C, so that only one pulse per switch operation is generated. In the W mode of S20, the latch is reset by the carry output pulse of IC4, which occurs only at the end of a word, stopping the clock at the end of one word. In the B mode of S20, the 110 baud clock is always off and clocking is provided only by S17, IC5C, and IC9C. Serial output is thus controlled by the rate that S17 is operated. IC2 and 3 are data switches that determine the output bit pattern. They are controlled by S19 and IC7B. In the rubout position of S19, both IC2 and IC3 are disabled, forcing all outputs low. In the "A" position, the Q output of IC7B is forced to a logic low so that the A outputs of IC2 and IC3 are selected. In the "B" position, the Q output is high and the B outputs are selected. In the "A/B" position of S19, IC7B is clocked by the carry output of U4. As was stated earlier, this occurs only at the end of a word. The Q output of IC7B thus goes from high to low on every other word, and A and B will alternate. IC4 is a binary counter that

has been programmed for a count of 11. It addresses the data selector, IC1, which takes the parallel data and provides serial output. The start bit and two stop bits have been hardwired at IC1. It should be noted that all parallel inputs to IC1 are inverted so that the serial output will be normal.

The output is bypassed to minimize noise and is buffered by IC5D. The TTL output can sink up to 16 mA. The RS-232C output is provided by the op amp, IC8. The output levels will depend on the ± supply inputs to IC8. They should be equal and from 5 to 15 volts at 10 mA. The logic supply should provide +5 volts regulated at .25A.

The model shown in photo 2 was point to point wired on a Vector number 3677-2 DIP plugboard. If you want to wire wrap, use the number 3682-2 plugboard. The physical arrangement of the completed project is shown in photo 3. The enclosure is an LMB Model W-1C utility case.

Appreciation is expressed to Fred LaPlante who took the pictures, supplied the test printer, and provided useful suggestions during the design of this unit. A double sided printed circuit board for this design (with plated through holes) will be available. Readers may write to the author for details.■

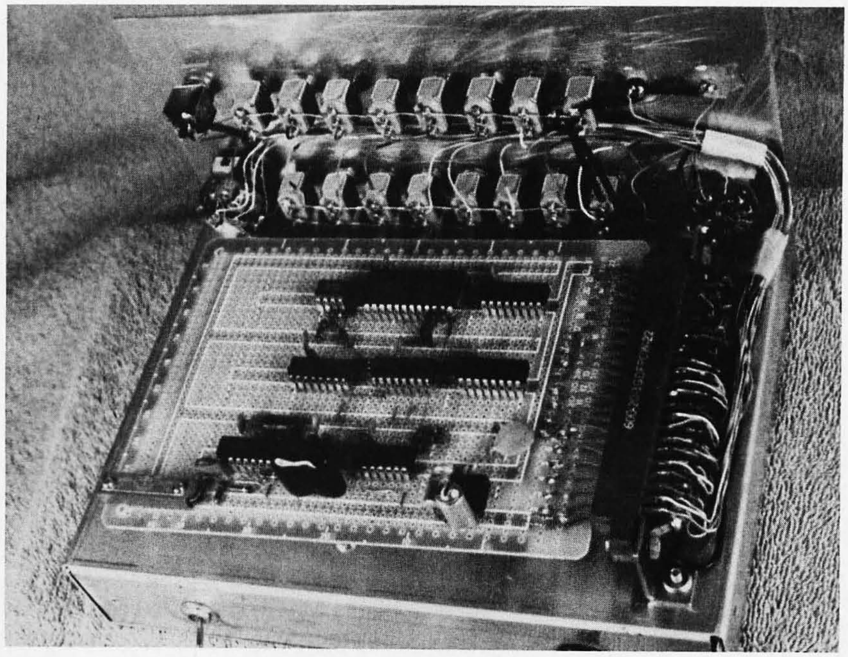


Photo 3: The general mechanical arrangement of the unit is shown in this picture. The front panel switches are toward the top of the photograph, with a wiring harness running to a socket for the plugboard.

Clubs and Newsletters



AEDS Convention

AEDS, the Association for Educational Data Processing, is holding its 14th Annual Convention in Phoenix AZ May 3-7 1976. For information contact Rick Meyer, convention chairperson, at Phoenix Union HS, 2526 W Osborn Rd, Phoenix AZ 85017, or call (602) 257-3045.

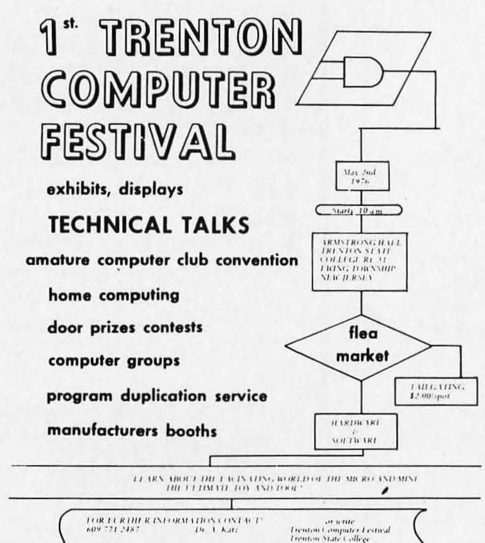
Amateur Computer Group of NJ

The February issue of the *ACGNJ News*, Volume 2 Number 2, reports on the January ACGNJ meeting, which included a demonstration of the new Intel SDK-8 8080 system design kit along with its documentation, courtesy of Art Chapman of Intel. The newsletter also included technical notes by Roger Amion (on memory expansion via piggyback mounting of ICs) and Art Chapman (a circuit to help program 2708 EROMs), and a compendium of Altair technical information from several sources.

The February *ACGNJ News* also announced the Trenton Computer Festival May 2 1976, beginning at 10 AM in Armstrong Hall at Trenton State College. The building is located in Ewing Township on Route 31. For detailed information, write:

Trenton Computer Festival
Trenton State College
Trenton NJ 08625

For direct contact, phone Dr A Katz at (609) 771-2487, or Sol Libes at (201) 889-2000, extension 248.



News from the Cleveland Digital Group

Gary Coleman of the Cleveland Digital Group writes that CDG has formed the Midwest Affiliation of Computer Clubs, its first goal being a conference for computer club members June 12 and 13. The conference will offer a place for manufacturers to show their wares and for computer hobbyists to visit and ask questions, etc.

For further information, contact Gary at 14058 Superior Rd, Apt 8, Cleveland OH 44118, or phone (216) 371-9304.

Dr Dobb's Journal of Tiny BASIC Calisthenics & Orthodontia

This is the journal of information about the PCC Tiny BASIC idea: interpretive language software developed by amateurs to implement such languages for microcomputers. The Volume 1 Number 1 issue contained 19 photocopied pages including reprints of the PCC articles on Tiny BASIC as a reference and starting point. Three issues are $3, with xerographic reproduction while circulation is low. Interested parties should write to Tiny BASIC & Orthodontia, Box 310, Menlo Park CA 94025.

The PILOT Information Exchange

PILOT is a language of Computer Aided Instruction (CAI) first created by John Starkweather at the University of California. According to an article by Sylvan Rubin which appeared in the November 1973 issue of *Computer Decisions*, the language is a simple structure oriented to the CAI situation, enabling one to construct interactive sequences of questions and answers which are presented to students.

The article by Dr Rubin and the Number 2 copy of the *PILOT Information Exchange Newsletter* were recently sent to BYTE. The newsletter is coordinated by Gregory Yob, and subscriptions are available at $2 per quantum of information. Contact:

Gregory Yob
c/o LO*OP Center
8099 La Plaza
Cotati CA 94928
(707) 795-0405

And From 65 Notes. . .

The Volume 3 Number 1 issue of *65 Notes*, publication of the HP-65 Users' Club, arrived at BYTE recently. This issue continues the trend of branching out toward the other programmable calculators. In Richard Vanderburgh's "SR-52 Notes," you'll find out about how to create several "pseudo codes," such as "Halt", branch to 000, and set error condition, for the SR-52 programs. These instructions, like their equivalents on

the HP-65, are not documented by the manufacturer and take trickery to accomplish.

This issue also includes several HP-65, HP-55, HP-25 and SR-52 programs. One of the most exciting is Richard Vanderburgh's SR-52 Assembler/Loader program which is probably the first self-assembler for a hand held calculator! It has some restrictions on its operations, but it is an enticing prospect. Also found in the mailing of the Volume 3 Number 1 issue is an index to *65 Notes* for July through December 1975, prepared by Alvin Gaines of Atlanta GA. If you want to find out what people are finding out about hand held programmable calculators, then you should investigate what *65 Notes* has to offer.

Contact Richard J Nelson, 2541 W Camden Pl, Santa Ana CA 92704, for a subscription at $12 per annum.

Lansing MI Activities

Computer Hobbyists Around Lansing is the name of the club which has been formed in that vicinity. If you wish to participate, contact one of the following people:

Joyce Church
Marvin Church
4307 Mar-Moor Dr
Lansing MI 48917
Phone: 482-9452

William Serviss
13121 Tucker Dr
DeWitt MI 48820
Phone: 669-3179

Daniel L Herrick
1214 Frederick Dr
PO Box 513
Owosso MI 48867
Phone: 723-3264

LO*OP Center

LO*OP Center, Inc, is an organization run by Liza Loop, dedicated to providing computers upon which children can exercise their minds. According to the brochure sent to BYTE:

> "Computers as learning and teaching media can be adapted to handle an infinite variety of subject material, including creative writing and forms of artistic expression . . . The staff concentrates on expanding the horizons of children through Computer Assisted Instruction. Special emphasis will be placed on material for young children which encourages creativity and the knowledge that they are in control of the machine. A large library of programs of interest to junior high and

> high school age people has already been developed in centers across the country and is available through the LO*OP center."

Activities at the LO*OP center will be of interest to the residents of Sonoma County CA and others at greater distance. In addition to the educational emphasis, the center has a weekly Teacher's Forum and is host to the Sonoma County Minicomputer Club meetings. Contact LO*OP Center, 8099 La Plaza, Cotati CA 94928 (707) 795-0405.

A Note on Dates and Deadlines

It takes time to put BYTE together. Material must be edited, typeset, proofread, pasted up, shot into negatives, printed, then mailed. Thus, notices should be in our hands no later than the 20th of the third month preceding publication. Deadline for the August 1976 issue is May 20 1976.

DACS Newsletter

The Volume 1 Number 4 issue of the *Denver Amateur Computer Society Newsletter*, dated February 1976, mentions meetings, classes for beginners, and the group's growth. A lecture by Adam Osborne, Osborne & Associates, arranged and scheduled on short notice February 5, was excellent, and well received by the DACS members, reported the newsletter. Also worth noting is the formation of special interest user groups for the various processors of members. Contact Denver Amateur Computer Society at PO Box 6338, Denver CO 80206.

Hoosier Amateur Computer & Kluge Society (HACKS)

Ray Borrill, who will soon be opening a computer store called "Data Domain," phoned to urge club activity in the vicinity of Bloomington IN. Interested individuals should contact Ray at 111 S College Av, Bloomington IN 47401.

Journal of Community Communications

Lee Felsenstein publishes the *Journal of Community Communications,* a compendium of personal opinions on the social implications of computers. *JCC* is published occasionally by LGC Engineering at $1 per issue, 12 issues for $10. The address is LGC Engineering, 1807 Delaware St, Berkeley CA 94703

The Analytical Engine

The Chesapeake Microcomputer Club is the result of activity in the Washington-Baltimore-Northern Virginia area, documented in *The Analytical Engine*. The Chesapeake Microcomputer Club is to be commended for thinking up an excellent and unique name for the newsletter, one which reflects the origins of computers. The Volume 1 Number 2 issue of *The Analytical Engine* includes a report from the Caretaker Board by Rich Kuzmack, an editorial by Philip N Hisley summarizing the growth of the club, a technical note by Jeff Schmitt on using FIFO memory chips (his example is the Fairchild 3341), a technical note by Alan Hastings showing how to program a software UART with detailed 8080 code for his version, announcement of a club visit to MOS Technology February 18-20, and biographies of the candidates for club officers along with an election ballot. Meetings have been held at "The Other Barn," Oakland Mills Village Center, Columbia MD. For information write Chesapeake Micro Computer, 236 St David Court, X4, Baltimore MD 21030. Editor of *The Analytical Engine* is Philip N Hisley, (301) 667-9690.

BYTE'S BUGS

Here lies documentation of known bugs detected in previous editions of BYTE . . .

RS-232 Levels

In Gary Liming's article, "Data Paths," in February 1976 BYTE, there is an error in the statement of the RS-232 voltage levels at the top of the first column on page 39. The RS-232 specification, according to a Xerox excerpt supplied by Ron Finger of Anchorage AK, provides that the mark (logical 1) state shall be a voltage more negative than −3 volts with respect to ground; that the space (logical 0) state shall be a voltage more positive than +3 volts with respect to ground. The maximum magnitude in either direction is specified as 25 volts with respect to ground.

Donald Zanolla of Burbank IL points out the following discrepancy in the sample code given on page 62 of BYTE January 1976, "Taking Advantage of Memory Address Space."

The symbol X could not be used in a program assembled by the Motorola 6800 assembler; the example would be correct if a different symbol such as "XX" were used in its place, ie:

```
LDX     XX      fetch first operand
                (16 bits);
STX     A16
         .
         .
         .
```

Connecticut Microists

George Ahmuty, 6011 Wendy Ln, Westport CT 06881, is interested in contacting other Connecticut "Microists" and forming a club. George can be reached by phone at 227-8534.

News of NECS

The March 3 meeting of the New England Computer Society, held at the Mitre Corporation's Cafeteria (Building C) in Bedford MA, 8 PM, heard a talk by Tom Miller, system architect for Texas Instruments, on the design of the TMS9900 processor. This microprocessor is scheduled to be available through TI distributors in May, with a price of $99.32 in quantities of one. (BYTE published Robert Baker's "Microprocessor Update" on the TMS9900 in the April 1976 issue.) The large quantity price strategy of TI, according to Mr Miller, was to make the TMS9900's price be about twice the price of an 8 bit processor in equivalent volume. The appearance of a TMS9900 based processor in the personal computing market cannot be far away, and Mr Miller strongly implied that TI is trying to encourage that particular use of their new processor. The talk's technical content included the machine's general architecture and a series of 10 to 15 slides on the practical details of TTL interfaces to the chip in small systems contexts.

Other activities were Dave LeVine's report on group purchase activities, including a one time purchase of IM6100 parts for club members at the 100 piece price, arranged with a local distributor. Dave Day presented an introductory session on hardware practices, oriented toward the members with no detailed hardware background. Doug Johnson presented a concurrent introduction to software concepts applicable to microcomputers.

Inquiries about NECS should be directed to PO Box 198, Bedford MA 01730.

Long Island Computer Association (LICA)

The Long Island Computer Association held its first meeting on January 16 1976 and formed a steering committee. The club is open to anyone, amateur or professional, with an interest in computers, applications, programming or related subjects. Meetings are held at the New York Institute of Technology, Building 500, Route 25A and Whiteney Ln, Old Westbury NY, usually on the third Friday of each month. For further information, contact Gerald Harrison, evenings until 9 PM at (516) 938-6769, or write Gerald at 36 Irene Ln E, Plainview NY 11803.

Kansas City Club?

Earl G Day, 13208 W 94th Ter, Lenexa KS 66215, would like to contact individuals interested in forming a computer club in the Kansas City area. Earl's phone number is (913) 492-9315.

Ithaca NY Computer Group

Steve Edelman, 204 Dryden Rd, Ithaca NY 14850, sent BYTE a note announcing formation of the Ithaca Computer Group. One of the first activities was to arrange a bulk purchase of 91L02A RAM chips for members. The Ithaca Computer Group meets "semi sporadically, but mostly on the second Sunday of the month." For information call Steve at (607) 272-2339.



TI SR-60 Desktop Programmable Calculator.

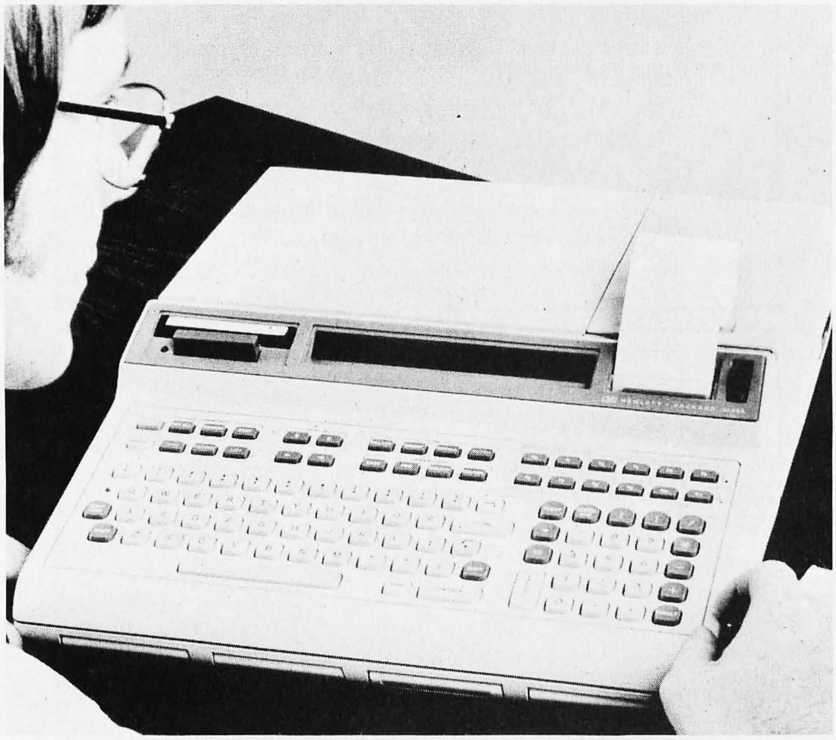
What's New?

Is This a Luxury Desk Top Version of the SR-52?

The only way to answer that question might be to go to a dealer or department store which handles the Texas Instruments calculator products line and compare. But the SR-60 gives the user much more in a desk top package, available at a suggested retail price of $1695. The extras include: a built in 20 character prompting display, a 20 character wide printer which can record the results of calculations or traces and dumps of programs in the machine, a standard memory of 480 program steps and 40 data locations, an option (at $700) to expand memory capacity to 1920 program steps and 100 data locations, algebraic expression notation with 9 levels of parentheses, up to 78 program labels, 15 user defined functions, 10 flags, 10 branch operations, 4 levels of subroutine stacking, 2 modes of indirect addressing, and a complete program editing capability using the display, with the option of listing programs on the thermal printer. The SR-60 calculator also features a trace option useful for debugging program operation. Mass storage for this calculator is provided by 2 inch by 10.5 inch (5.08 cm by 26.67 cm) magnetic cards.

An extensive library of prerecorded programs is available for financial, mathematical, electrical engineering, statistics and surveying. The unit comes with a basic library of 10 prerecorded programs, operating manual, programming manual, programming pad, blank magnetic cards, and card holders. Inquiries should be directed to Texas Instruments, Inc, PO Box 5012 MS84, Dallas TX 75222, Attn: SR-60. The SR-60 will be available through TI office equipment dealers and department stores.■

HP 9825A Desktop Programmable Calculator.



A Desk Top High Level Language Machine

Hewlett-Packard's 9825A Desk Top Programmable Calculator is an option which might prove quite attractive to individuals wishing to tradeoff dollars against the time and trouble of kit oriented approaches. At $5900 for the basic desk top unit the user will find a "black box" which is programmable in a high level language called HPL, a 32 character alphanumeric display and matching 16 character wide alphanumeric printer for examining programs and results of programs, 12 user programmable keys which through use of a shift key provide 24 user programmable program options, and a tape cartridge device which can be used for off line storage of up to 250,000 bytes transferred at a data rate of 2,750 bytes per second.

The standard unit has 8 K bytes of internal programmable memory, which is expandable to a maximum of 32 K. Optional plug in ROM packets are available to provide such extended HPL functions as string handling, general language extensions such as

FOR/NEXT loops, matrix operations, plotter control, general IO, extended IO, etc. Up to four of these ROM packets can be plugged into the machine at one time.

The other aspect well worth considering for experimentalists is the fact that the machine is intended to be used with IO for process control, laboratory instrument control, and other nonstandard IO functions. By implication this means that the home experimenter with a large budget could find this machine to be an excellent controller of such items as burglar alarms, fire alarms, kitchen information systems, model railroad layouts, etc. A business person could interface the HP printer for this machine (at extra cost) and develop packages which could handle most small business data processing needs, using the tape cartridge IO for mass storage of statistics and records (with hard copy back up for the conservatives in the audience).■

CPU Emulator for 6800-Based Microprocessor Systems

The DICE/68 is a microcomputer system development aid designed to provide users of the 6800 microprocessor and the Motorola Exorciser development system with the capability of in-circuit CPU emulation. Other features of this design and debugging tool include status indicators for data and address buses, plus additional system control features.

By plugging a 40 pin DICE/68 adapter directly into the 6800 CPU socket on one's own hardware, it is possible to debug a 6800 system prototype, complete with RAM, ROM, IO circuitry and two phase clock, using the full range of diagnostic aids available through the Motorola EXBUG™ operating system found in its Exorciser product. The user can, after specifying through DICE/68 the block of memory allocated to the prototype system, begin the hardware debugging phase of development.

The DICE/68 system, in addition to being a good microprocessor system debugging aid, can also be used effectively in a production testing environment. This product will be of primary interest to BYTE readers engaged in microprocessor systems design and microprocessor system service activities. The price is $795 with delivery 2 to 4 weeks ARO.

For further information contact Digital Electronics Corporation, 2126 Sixth St, Berkeley CA 94710.

"Chip" Off the Olde PDP 8/E:

The Intersil IM6100 Part I

If you are looking for an answer to the software availability problem, this microprocessor may be it.

Robert Nelson
Chief Engineer
PCM Inc
180 Thorup Ln
San Ramon CA 94583

Of all the computers in the world today, there are probably more Digital Equipment Corporation (DEC) PDP-8s than any other kind.* A recent article in Electronic Engineering Times (October 20, 1975, page 2) estimates "... there are upwards of 70,000 PDP-8s in use in the field, and that 60 percent of all recent electrical engineering graduates have been exposed to the PDP-8 and its software." The PDP-8 at this point may truly be *the* universal computer. The latest version from DEC is the PDP-8/E.

There are some good reasons for the PDP-8's popularity. It has been available since 1964. It has always been relatively inexpensive (at least for industrial users). It employs a convenient parallel word length of 12 bits, which can make it more powerful than a 8 bit machine. An important consideration for the hobbyist is that it is easy to understand, both hardware and software. And there is a literal flood of software available for it. DEC itself makes available more than a *thousand* fully developed and documented programs for the 8. DECUS, a DEC-sponsored non-profit, free-membership user's group maintains a library of hundreds of other programs. And many users, such as universities, US government operations and research groups have developed and made available additional programming for the 8.

Just a few of the fully-documented programs available from DEC for the PDP-8:

- PAL III – A basic 2- (or optionally 3-) pass assembler.
- MACRO-8 – An advanced assembler with all the features of PAL III plus many others, such as user-defined macros, double-precision integers, floating-point constants, Boolean algebra operators, etc.
- FORTRAN IV compiler – Runs in 8K of memory.
- Extended BASIC translator – Also for 8K of memory.
- FOCAL – An interactive language similar to BASIC and FORTRAN, but requiring less memory than either, and even easier to learn.
- DIBOL – A business-oriented language similar to COBOL.
- TECO – A text editor and corrector.
- DDT – A program debugging routine.

There are also many diversionary, game (yes, including Space War) and educational programs available.

In addition to software, DEC makes available many tutorial handbooks for the PDP-8 machines. The *PDP-8/E Small Com-*

PCM manufactures the PCM-12, a machine based on the IM6100, which is much like a kit-form PDP-8/E.

The PDP-8 at this point may truly be *the* universal computer.

Since the IM6100 features *static operation*, the basic CPU clock can be varied from 0 Hz to the maximum operating speed. A speed control could be added to an IM6100 based processor.

puter Handbook and *Introduction to Programming* cover the PDP-8 hardware and how-to methods for PDP-8 software, respectively. They are superbly written, assume you start reading with almost no knowledge of computers, and use absolutely no "computerese" that isn't fully explained first. For the tyro wanting to learn basic computer operation and programming, these inexpensive paperbacks are hard to beat. Write to DEC, 146 Main St, Maynard MA 01754, or phone (617) 897-5111.

Now with this brief introduction to the PDP-8, it is easy to see that it might make an ideal machine for the computer hobbyist to own. However, the prices are still a bit steep for home use – several thousand dollars for an operating machine. But now Intersil (10900 N Tantau Av, Cupertino CA 95014; phone (408) 996-5000) has introduced the IM6100, a 40 pin DIP microprocessor chip that recognizes the PDP-8/E instruction set and can therefore execute PDP-8 software. That means that if you build yourself a small computer around the 6100, it can immediately become a working machine, rather than simply an empty brain, by filling its memory with almost any program that will run on the PDP-8. And that includes BASIC, FORTRAN, FOCAL, assemblers, editors, games, etc.

The bus structure of the IM6100 can easily be adapted to provide a subset of the PDP-8 OMNIBUS signals. Therefore all programmed IO interfaces for the PDP-8 (Teletype, paper tape reader, punch, printer, etc.) will operate with the IM6100 without any hardware or software modification.

The essential differences between the 6100 and PDP-8/E CPU are few. The 6100, unlike the PDP-8/E, does not provide for timesharing (at least not yet, but see the discussion of Intersil's support chips for the IM6100 in the second part of this article). The 6100 does not support the DEC extended arithmetic element (EAE). This is an optional piece of hardware that speeds execution of high level mathematical routines, such as floating-point arithmetic, trig and log functions, etc. (Of course the 6100 can, with standard PDP-8 software, execute any of these mathematical functions without the EAE; it just calculates the results in software, more slowly.) The direct memory access (DMA) structure of the IM6100 differs from that of the PDP-8/E. The IM6100, having a limited number of pins, does not provide continuous or "real time" access to all its internal registers (Accumulator, MQ Register, Link, etc.). This "fault" the 6100 has in common with all microprocessors and requires that the front panel of a small computer built around it be implemented in software. More about that later.

Some of the features of the IM6100 are:

- The 6100 contains six 12 bit registers, a programmed logic array (PLA), arithmetic logic unit (ALU) and all the necessary gating and timing circuitry to implement a complete PDP-8/E central processor unit (CPU).
- Silicon gate CMOS construction for low power, single supply operation. Silicon gate means that chip size is small for CMOS, and that leads to lower chip cost. CMOS construction means excellent noise immunity.
- The chip needs just a single 5 to 11 V power supply, and it doesn't need to be regulated. Current drain at 5 V is about 4 mA.
- All inputs and outputs are fully TTL compatible, when operated at 5 V.
- Static operation (a rare feature in a MOS processor). All registers inside the chip are static, which means you can shut off the clock without losing data. This makes it possible to put both single clock or single instruction

As in many of the 40 pin packaged 16 bit microcomputers, the IM6100 uses a common 12 bit data bus for both address and data information.

***The following are registered trademarks of Digital Equipment Corporation, Maynard MA: DEC, PDP, FOCAL, OMNIBUS.**

Memory address space in a PDP-8 (IM6100) is a collection of fields of pages of memory locations.

buttons on the front panel, a great aid for program debugging.

- On chip crystal oscillator. Just put a crystal across pins 14 and 15 and the chip will generate all its own timing. Or substitute a TTL pulse generator at pin 14 and clock the chip at any speed from 0 Hz to the maximum allowable clock frequency.
- Interfaces directly with standard solid state programmable random access memories (2102s, for example), PROMs and ROMs, as well as standard TTL memories and logic.
- Operating at 10 volts, with an 8 MHz crystal, the 6100 will do a memory to accumulator binary addition in just 2.5 μS. This spec makes it the fastest available MOS microprocessor. (Hobbyist operation will probably be at 5 volts, though, to make TTL interfacing easy. This reduces the maximum clock frequency to 4 MHz and increases the above-mentioned add time to 5 μS. Still, that's one of the fastest chips around when you consider that this is a 12 bit addition.)

A Lap Around the Pins

A basic understanding of the operation of the IM6100 can begin by familiarizing oneself with the operation of each of the pins on the chip:

Pin 1: Supply voltage. Typically 5 V at about 4 mA.

Pin 2: RUN line. When this line is high, the machine is running. This pin operates in conjunction with pin 6, the RUN/HALT line. Negative pulses on the latter pin cause the 6100 to alternately go to the run and halt states.

Pin 3: DMAGNT line. When a DMA request is generated, by a low level on the DMAREQ line (pin 4), the 6100 grants the request at the end of the current instruction by presenting a high level on this line.

Pin 4: DMAREQ line. See pin 3.

Pin 5: CPREQ line. A low level presented to this line causes a control panel interrupt to occur after completion of the current instruction.

Pin 6: RUN/HALT line. See pin 2.

Pin 7: RESET line. A low level presented to this pin clears the 6100 accumulator, loads 7777_8 into the program counter and puts the CPU into the "halt" state (RUN line low).

Pin 8: INTREQ line. A peripheral device requests an interrupt by presenting a low level to this line.

Pin 9: XTA line. This timing line goes high once each machine cycle. It is typically used by external logic to indicate the "read" portion of the cycle. See the timing diagram of figure 1.

Pin 10: LXMAR line. The LXMAR pulse is developed by the CPU once each cycle. It is primarily used externally to latch the address sent out on the 12 data lines (DX0 to DX11) by the 6100 at T1 time.

Pin 11: WAIT line. The 6100 samples the WAIT line at T3 time (read) and T6 time (write). If it finds the WAIT line low, it extends the current state by increments of the clock period until WAIT goes high again. By using this feature, the 6100 can operate with memories of any speed.

Pin 12: XTB line. Similar to XTA; another external timing line that can be used to drive external devices. It is high only during T1 and T6.

Pin 13: XTC line. The most important timing line. High through T3, then goes low for the rest of the cycle.

Pin 14: Crystal input for internal oscillator, or drive point for an external clock oscillator.

Pin 15: The other side of the crystal, or grounded if an external oscillator is used.

Pins 16 to 28 (except 26): DX0 to DX11. These are the 12 multiplexed, bidirectional lines that carry instructions and data into and out of the chip. DX0 is the most significant bit (MSB), DX11 the least significant bit (LSB). (This is the opposite of numbering conventions used with most 8 bit processors.)

Pin 26: Ground.

Pin 29: Link line. This line, when high, indicates that the Link flip flop is set.

Pin 30: DEVSEL line. The IM6100 employs four select lines to distinguish cycles involving main memory, external devices, control panel and the switch register from

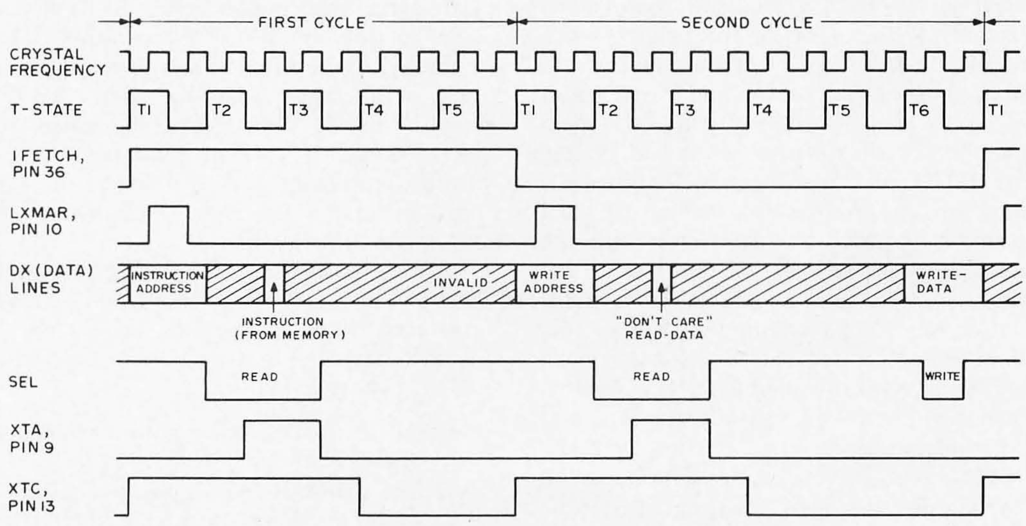


Figure 1: IM6100 CPU Timing Diagram. This is an example of the timing of several IM6100 processor signals during the execution of a DCA instruction.

one another. The DEVSEL line, when low, asserts that the cycle involves an external device.

Pin 31: **SWSEL line.** This select line, when low, indicates that the switch register is to be used for the read operation.

Pin 32: **C0 line.** This line, like the C1 and C2 lines, is used by external devices to control the operation of the CPU during an IOT instruction (see discussion of instruction set, below).

Pin 33: **C1 line.** See pin 32.

Pin 34: **C2 line.** See pin 32.

Pin 35: **SKP line.** Similar to the C lines, above. When the external device asserts this line low during an IOT instruction, it causes the 6100 to skip the next sequential instruction.

Pin 36: **IFETCH line.** This line is high throughout each cycle that is used to fetch an instruction.

Pin 37: **MEMSEL line.** This select line, when low, indicates the cycle involves main memory action.

Pin 38: **CPSEL line.** When low, this select line indicates the instruction or data is to be read from or written into the control panel memory.

Pin 39: **INTGNT line.** This line goes high when the 6100 grants a device interrupt.

Pin 40: **DATAF line.** This line goes high during the execute phase of an indirectly addressed AND, TAD, ISZ or DCA instruction. It allows the extended address element to select a different field than that from which the instruction itself was taken.

Timing Diagram

IM6100 timing is quite easy to understand. Refer to the basic CPU timing diagram shown in figure 1. This diagram is given in the form of an example — the complete fetch and execute cycles of a DCA instruction. This instruction deposits the 6100 accumulator contents into a selected memory location, then clears the accumulator.

T-state timing is derived directly from the crystal. Each state requires two complete cycles of the crystal oscillator. Then each machine cycle requires either five or six T-states, depending on the instruction. Six T-states are required when the particular machine cycle involves a write operation. Most instructions require two or three machine cycles to complete their fetch and execute phases, but a few require four cycles.

The example begins with the first cycle, fetching the DCA instruction from memory. Throughout the fetch cycle the IFETCH line stays high. During T1 the instruction address, derived from the IM6100 program counter, is put on the data lines (DX). The LXMAR pulse is then used to clock this address into a 12 bit wide latch in the memory. (The trailing edge of LXMAR should be used to produce this latching action, so the DX data has plenty of time to settle at the latch inputs.) The memory then retains this address until another LXMAR pulse comes along in the next cycle.

The next thing that happens is that one

of the select lines, MEMSEL, DEVSEL, CPSEL or SWSEL, goes low at the rising edge of T2. The purpose of the select is to specify whether the action during the cycle concerns main memory (MEMSEL), an external device (DEVSEL), the control panel memory (CPSEL), or the front panel switch register (SWSEL). In our example we are first fetching an instruction from main memory, then depositing data into main memory, so in both cycles it will be the MEMSEL line that will go low. This signal is used (in the first cycle) to allow the main memory to drive the data lines, with the data to be received by the IM6100. In the first cycle of the example the data transmitted by memory will be the DCA instruction. States T3, T4, and T5 of the first cycle are then used by the 6100 for internal operations such as register transfers and ALU operations.

The second cycle starts with the 6100 putting the address to be written into on the data lines, and outputting the LXMAR pulse to latch this address into main memory. (This address is a part of the DCA instruction which was fetched in the first cycle.) Although the second cycle is a "write" cycle, the 6100 will first perform a dummy (or "don't care") read at T2 when the MEMSEL line is driven low. This read data is ignored by the 6100. States T3, T4, and T5 are again used for internal operations. At T6 the 6100 puts the data to be written into the latched address in main memory onto the data lines. Then the MEMSEL line is driven low to actuate the write operation. The memory itself differentiates between read and write, when MEMSEL goes low, by monitoring the XTA or XTC line. When the X line is high it is a read, when low a write. (XTC is probably the best line to use for this, because it provides a little hold time after a read operation.)

This completes our walk through a typical instruction fetch and execution. Some instructions take more cycles, but the timing is still basically the same. Here are some things to keep in mind:

1. All instructions start with a five state fetch cycle.
2. The LXMAR pulse occurs in every cycle. In addition to latching addresses, it can be used to clear or set flip flops, etc., in peripheral devices.
3. In a given cycle, only one select line operates. However, some instructions involve a cycle which uses one select line, followed by a cycle which uses another select line. For example, see the discussion of the IM6100's unique control panel provisions, below.
4. DX line data moves in both directions, to and from the 6100. Only in *some* T-states is this data valid; in other states these lines are in the high impedance state and just float. See the timing diagram, figure 1.
5. The XTA, XTB and XTC lines serve as indicators of the current state of the 6100, within the cycle.

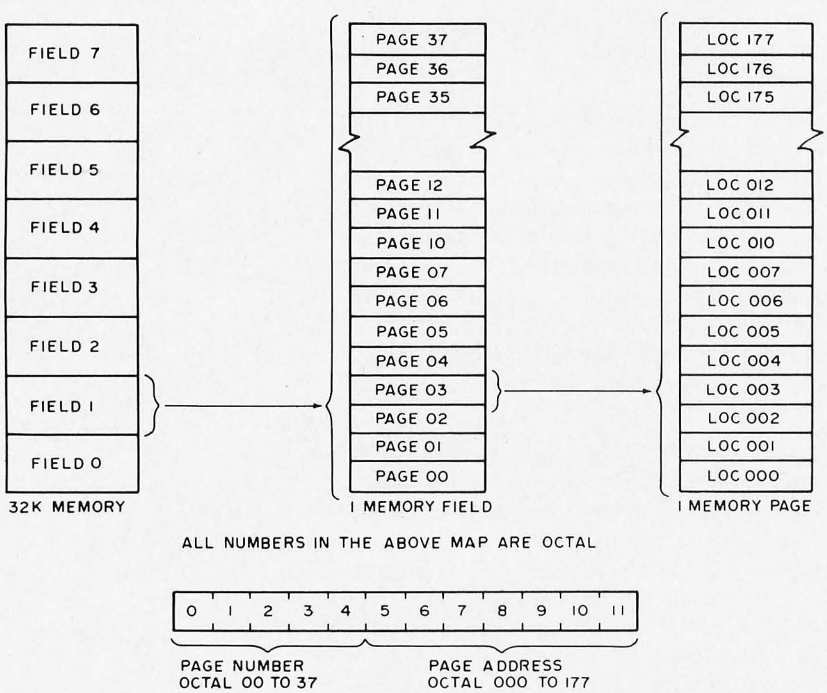


Figure 2: Memory Organization. The memory address space of the IM6100 (and the PDP-8) is divided into fields of 4096 words. Each field in turn is considered to have 32 pages. Within each page there are 128 memory locations. The division into fields, pages and locations is required by the way in which the instructions reference memory.

Instruction Set

The IM6100 instruction set is identical to that of the PDP-8/E. All instructions are 12 bits long, so it always takes just one machine cycle to fetch an instruction. The 6100 makes no distinction between instructions and data; it can manipulate instructions as data or execute data as instructions when it is programmed to do so. Software persons will recognize this convenience. The instruction can be divided into three categories: memory reference instructions, operate instructions and input output transfer (IOT) instructions. But before discussing these three instruction types, let's get acquainted with the memory structure to be used with the 6100 (again, it's just like the PDP-8).

Like the PDP-8, the IM6100 has a basic addressing capacity of 4096 (4 K) 12 bit words. This addressing capacity is a natural

result of the 12 bit word width, and can be expanded to 32 K, or beyond, by addition of a simple extended address element module. This module, when designed for PDP-8 software compatibility, requires about 25 SSI and MSI TTL chips (but see discussion of Intersil's IM6100 LSI support chips, to follow in part 2 of this article). A maximum memory size of 32 K can be implemented with a PDP-8.

The memory system is organized into 4096-word blocks called "fields." The first 4 K words are in field 0. If a full 32 K of memory is installed, the uppermost memory field is numbered 7. In any given memory field every location has a unique 4-digit octal (12 bit binary) address, 0000 to 7777 decimal 0000 to 4096. Each memory field is further subdivided into 32 pages of 128 words each. Memory pages are numbered sequentially from octal 00, containing octal addresses 0000 – 0177, to octal 37, containing octal addresses 7600 – 7777. The first five bits of a 12 bit memory address denote the page number and the low order 7 bits specify the address of the memory location within the given page, called the page address. See figure 2.

During an instruction fetch cycle, the IM6100 fetches the instruction pointed to by the program counter (PC). The contents of the PC are transferred to the memory address register (MAR), and the PC is incremented by one. The PC then contains the address of the next sequential instruction. The MAR contains the address of the "current" instruction, which must be fetched from memory. Bits 0 – 4 of the MAR identify the current page, that is, the page from which instructions are currently being fetched, and bits 5 – 11 identify the location within the current page.

The memory reference instructions operate on the contents of a memory location or use the contents of a memory location to operate on the accumulator or program counter. The first three bits of a memory reference instruction specify the operation code, or "opcode," and the low order 9 bits the operand address, as shown in figure 3.

Bits 5 – 11, the page address, identify the location of the operand on a given page, but they do not identify the page itself. The page is identified by the page bit, bit 4. If bit 4 is a 0, the page address specified is interpreted to be on page 0. If bit 4 is a 1, the page address is interpreted to be a location on the current page, that is, the page from which the current instruction was fetched.

For example, if bits 5 through 11 represent octal 023 and bit 4 is a "0", the location referenced is the absolute octal address 0023. However, if bit 4 is a "1" and the current instruction was fetched from octal location 4610, the page address 023 designates the absolute octal address 4623.

By this method 256 locations may be directly addressed, 128 on page 0 and 128 on the current page. Other locations are addressed by utilizing bit 3. When bit 3 is a "0", the operand address is a direct address. But when bit 3 is a "1", the address is taken to be "indirect." An indirect address (or "pointer" address) identifies the memory location that *contains* the desired absolute address. To address a location that is not directly addressable (not on page 0 or the current page), the absolute address of the desired location is stored in one of the 256 directly addressable locations. This directly addressable location is used as the "pointer." (To make life simpler, remember that an indirect address and pointer address really mean the same thing – the hiding place that contains the full 12 bit effective address of the operand you *really* want.)

So the IM6100 has direct and indirect addressing capability. It has one other addressing ability, autoindexing. Octal locations 0010 to 0017 in page 0 are autoindexed. If these locations are addressed indirectly, the contents are incremented by one and restored before they are used as the operand address. These locations may, therefore, be used for indexing applications.

The memory reference instruction mnemonics, and their opcodes, are shown in Table 1, along with an explanation of what they do and how long they take to execute.

The second category of instructions is termed the operate instructions, all of which have an opcode of octal 7. These instructions are all used for IM6100 internal operations, such as conditional and unconditional skips, accumulator rotates (either left or right, and one- or two-bit shifts), clearing and setting the accumulator and

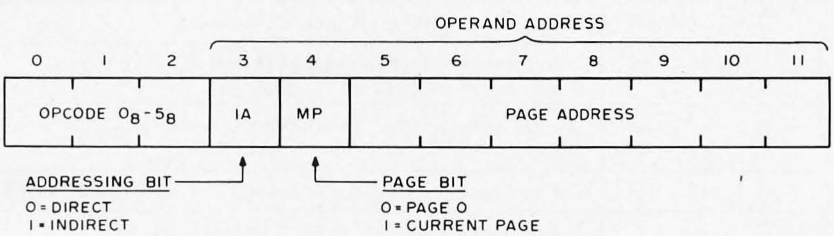


Figure 3: Memory Reference Instruction Format. Memory reference instructions make use of the memory organization concepts in figure 2. The page address identifies the particular word desired; the page bit (MP) selects whether the current page or page 0 is selected; the addressing mode bit allows an additional single level of indirection so that data outside the current page (but still within the current memory field) can be referenced.

Mnemonic	Binary Op Code	Operation	Number of T-States Required (direct addressing)
AND	000	Logical AND – The memory location addressed is AND'ed with the AC. Result remains in AC.	10
TAD	001	Binary ADD – Memory contents are added to AC; result remains in AC. Carry complements Link.	10
ISZ	010	Increment and skip if zero – The memory location is incremented. If the result is zero, the next sequential instruction is skipped.	16
DCA	011	Deposit, and clear AC – The contents of the AC are deposited in the addressed memory location, then the AC is cleared.	11
JMS	100	Jump to Subroutine – The PC contents are stored in the addressed memory location. The PC is then set to one address higher than that in the instruction.	11
JMP	101	Unconditional Jump – The next instruction is taken from the address embedded in the current instruction.	10

Table 1: Summary of Memory Reference Instructions. This set of six instructions references memory in the format of figure 3. Instructions with binary opcodes 110 and 111 are the IOT and Operate instructions, respectively.

Control Lines C0	C1	C2	Operation
H	H	H	The content of the AC is sent to the device.
L	H	H	The content of the AC is sent to the device. Then the AC is cleared.
H	L	H	The device data is OR'ed with the AC; result remains in the AC.
L	L	H	The device data is loaded into the AC.
X	H	L	The device data is added to the contents of the PC.
X	L	L	The device data is loaded into the PC.

X – don't care

Table 2: Input Output Control Lines. The states of the three control lines (defined by the IO hardware) tell the IM6100 what to do during an IOT instruction.

link, fetching the MQ Register to the accumulator, etc. These instructions use bits 3 to 11 in the instruction word (after the opcode of 7 in bits 0 – 2) to specify the exact operation to be performed. All these bits are available, of course, since all the operations specified are internal to the IM6100 itself and do not require specification of a memory address.

No detailed listing of these instructions will be given here since it is a lengthy list, and it is clearly explained in the IM6100 data sheet (and also in DEC's *Small Computer Handbook*). However, it should be pointed out that these instructions are termed microinstructions by DEC, since they can be combined by setting or clearing the proper bits in the instruction word. This often cuts down the number of individual steps necessary in a program. It is possible, for example, to use a single instruction to produce CLL followed by RTL, which will clear the Link *then* rotate the accumulator two positions to the left.

The third category of instructions consists of the input output transfer (IOT) instructions. These all have an opcode of octal 6 and are used to initiate the operation of peripheral devices and to transfer data between peripherals and the 6100. (Actually we are talking here about *programmed* data transfers; data can also be transferred to or from peripherals by means of interrupts and direct memory access, to be explained later.)

In an IOT instruction, bits 0 to 2 are always set to binary 110. Bits 3 to 8 are the device selection code, used to select the peripheral device, and bits 9 to 11 specify the operation to be performed with the selected peripheral. The device selection code octal 00 in bits 3 to 8 is reserved for processor IOTs. There are eight of these: octal 6000 – 6007. They are used by the CPU for certain housekeeping operations such as turning on and off the interrupt system, fetching flag bits to the accumulator, etc. These are explained in detail in the 6100 data sheet.

A programmed data transfer begins when the IM6100 fetches an instruction from memory and recognizes it as an IOT instruction. The 6100 sequences the IOT instruction through a 2 cycle execute phase referred to as IOTA and IOTB. The instruction must be latched into the external device, using the LXMAR pulse. DEVSEL is the active select line for all IOT instructions. The selected peripheral device controls the IM6100 during the data transfer by means of the C0, C1, C2 and SKP lines. The type of data transfer is specified by the peripheral

device by asserting the control lines as shown in table 2.

The SKP line, when asserted low by the peripheral device during an IOT, causes the IM6100 to skip the next sequential instruction. This feature is used to sense the status of various flags in the device interface. The C0, C1 and C2 lines are treated independently of the SKP line.

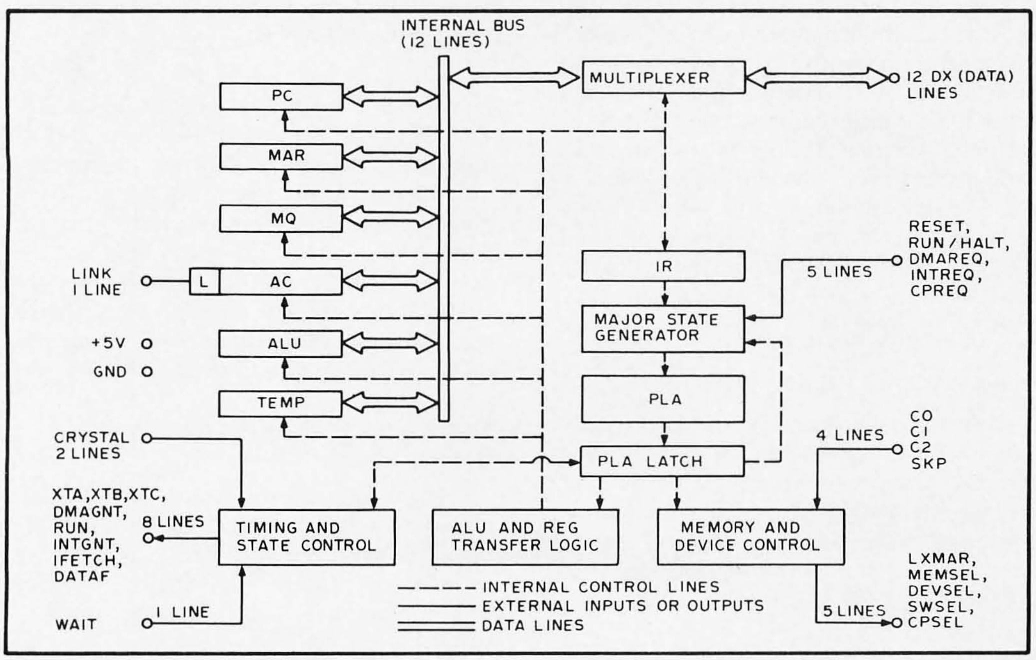
Except for processor IOTs, all IOT instructions are nonspecific in that, unlike all other instructions, the operation that they perform is not known by the CPU. Rather, the hardware designer specifies what each of these instructions does by the logic he builds into the interface for the specific peripheral device. The IOT instructions work in conjunction with the C0, C1, C2 and SKP lines on the 6100 chip. Let's take an example: For a PDP-8 compatible Teletype interface, it is necessary that the IOT instruction octal 6034 cause the TTY keyboard data to be "OR'ed" into the 6100 accumulator. Referring to table 2, it is seen that in order to cause device data to be "OR'ed" into the accumulator, it is necessary to pull control line C1 low while C0 and C2 remain high. The interface logic, then, must recognize the arrival of the octal 6034 code and assert C1 low. Similarly, IOT instruction octal 6031 must cause the next instruction to be skipped if the keyboard data ready flag is set in the device interface. To accomplish this, the interface logic must, upon arrival of the octal 6031 instruction code, test the data ready flag, and then if (and only if) it is set, assert the SKP line low. (By the way, in the typical DEC PDP-8 system, device code octal 03 has been assigned to the TTY keyboard and octal 04 to the TTY printer.)

It is seen, then, that the system designer has nearly complete freedom with the IOT instructions. He first decides what he wants a given IOT instruction to do, then builds the necessary "interpretive" logic into his peripheral interface. Those of us who want DEC software compatibility, though, must use the IOT instructions as already defined in the aforementioned *Small Computer Handbook* and *Introduction to Programming*, and make sure that our interface logic "understands" those instructions.■

This article is being printed in two parts. The second part continues the discussion, covering topics including interrupts, direct memory access, control panel software and support devices for the IM6100.

ORGANIZATION OF THE IM6100

Many of the microprocessor chips available today were not designed to be the heart of a general purpose minicomputer. They were primarily designed with dedicated industrial control applications in mind. But the Intersil IM6100 is an exception, since it imitates so well the structure of the PDP-8/E CPU. A block diagram of the 6100 is shown below.



Accumulator (AC)

The AC is a 12 bit register with which arithmetic and logical operations are performed. Data words may be fetched from memory to the AC or stored from the AC into the memory. Arithmetic and logical operations involve two operands, one held in

the AC and the other fetched from memory. The result of the operation is left in the AC. The AC may be cleared, complemented, tested, incremented or rotated under program control. The AC also serves as an input output register. All programmed data transfers pass through the AC.

Link

The Link is a 1 bit flip flop that serves as a high order extension of the AC. It is used as a carry flip flop for 2's complement arithmetic. A carry out of the ALU complements the Link. The Link can be cleared, set, complemented and tested under program control, and rotated as part of the AC.

MQ Register (MQ)

The MQ is a 12 bit register which is program accessible. The contents of the AC may be transferred to the MQ for temporary storage. The MQ can be OR'ed with the AC and the result stored in the AC. The contents of the AC and the MQ may also be exchanged.

Program Counter (PC)

The 12 bit PC contains the address of the memory location from which the next instruction is fetched. During an instruction fetch, the PC is transferred to the MAR and the PC is then incremented by one. When there is a branch to another address in memory, the branch address is transferred into the PC. Branching normally takes place under program control. However, during an input output (IOT) operation, a device may specify a branch address. A skip (SKP) instruction increments the PC by one, thus causing the next instruction to be skipped. The SKP instruction may be unconditional or conditional on the state of the AC and the Link. During an IOT operation, a device can also cause a SKP, by asserting the SKP line (pin 35) low.

Memory Address Register (MAR)

While accessing memory, the 12 bit MAR contains the address of the memory location that is currently selected for reading or writing. The MAR is also used as an internal register for microprogram control during data transfers to and from memory and peripherals.

Arithmetic and Logic Unit (ALU)

The ALU performs both arithmetic and logical operations, including 2's complement binary addition, AND, OR and complement. The ALU can perform a single position shift to either left or right, as well as a two bit shift in either direction. The ALU can also shift by three positions to implement a byte swap in two steps. The AC is always one input to the ALU. However, under internal microprogram control, the AC may be gated off and all ones or all zeros gated in. The second input to the ALU can be any one of the other registers under microprogram control.

Temporary Register (TEMP)

The 12 bit TEMP latches the result of an ALU operation before it is sent to the destination register, to avoid race conditions. The TEMP is also used as an internal register for microprogram control.

Instruction Register (IR)

During an instruction fetch, the 12 bit IR contains the instruction that is to be executed by the CPU. The IR specifies the initial step of the microprogram sequence for each instruction and is also used as an internal register to store temporary data for microprogram control.

Multiplexer (DX)

The 12 bit input output multiplexer handles data, address and instruction transfers into and out of the CPU, from or into the main memory and peripheral devices on a time-multiplexed basis.

Major State Generator and Programmed Logic Array (PLA)

During an instruction fetch the instruction to be executed is loaded into the IR. The PLA is then used for the correct sequencing of the CPU for the appropriate instruction. After an instruction is completely sequenced, the major state generator scans the internal priority network. The state of the priority network decides whether the machine is going to fetch the next instruction in sequence or service one of the external request lines.

Memory and Device Control, ALU and Register Transfer Logic

The memory and device control unit provides external select signals to communicate with peripheral devices (DEVSEL), switch register (SWSEL), memory (MEMSEL) and control panel memory (CPSEL). During IOT instructions this unit also modifies the PLA outputs, depending on the states of the four device control lines (SKP, C0, C1 and C2). The ALU and register transfer logic provides the control signals for the internal register transfers and ALU operation. ■

GODBOUT

BILL GODBOUT ELECTRONICS
BOX 2355, OAKLAND AIRPORT, CA 94614

YOU MAY PLACE MASTERCHARGE® OR BANKAMERICARD® ORDERS BY CALLING, 24 HOURS A DAY,

(415) 357-7007

TERMS: ADD 50¢ TO ORDERS UNDER $10. ADD SHIPPING WHERE INDICATED; OTHERWISE, ITEMS SHIPPED POSTPAID. CAL RES ADD TAX. NO COD---IT'S TOO MUCH PAPERWORK!

Our New 4k X 8 Ram Kit

ECONORAM™ $89.95

OTHER MANUFACTURERS HAVE PICKED UP ON OUR BUFFERED ADDRESSES AND DATA OUTPUTS...AND THEY'VE STARTED USING HYSTERESIS ON THE DATA IN, TOO...AND STATIC OPERATION...BUT WE'VE GOT THE RECIPE ON KEEPING THE PRICE DOWN. ALTAIR 8800 PLUG-IN COMPATIBLE. ONBOARD REGULATION, SOCKETS FOR ALL ICs, LOTS OF BYPASSING, 1A POWER CONSUMPTION, INSTRUCTIONS: THE ONLY THING WE SKIMPED ON WAS THE PRICE.

A 4k X 8 Erom Board Kit

ECONOROM™ $159.90

HERE'S A HOME FOR YOUR SOFTWARE, IN A MEMORY THAT DOESN'T FORGET. OUR EROMs MAY BE CHANGED AT WILL, SO YOUR BOARD WON'T BECOME OBSOLETE AS YOUR SOFTWARE CHANGES. EXPANDABLE TO 8K x 8. PROGRAM IT YOURSELF...HAVE US PROGRAM IT FOR A NOMINAL FEE... OR SPECIFY ONE OF OUR PRE-PROGRAMMED BOARDS: THEY COME WITH 8080 EDITOR, ASSEMBLER, AND MONITOR PRE PROGRAMMED. ALTAIR 8800 PLUG-IN COMPATIBLE.

Our Bargain Ram Kit

NAKED RAM 4k x 8 $79.95

THIS GENERAL PURPOSE BOARD IS DESIGNED SPECIFICALLY FOR JOLT SYSTEMS, BUT IS EQUALLY APPLICABLE TO OTHER BI-DIRECTIONAL BUSS SYSTEMS. IF YOU DON'T NEED THE ON BOARD REGULATION OR BUFFERS OF OUR "BIGGER BROTHER" 4K x 8 BOARD, THEN THIS IS THE WAY TO IMPLEMENT COST-EFFECTIVE MEMORY. LOW POWER --- UNDER 750 MA. SAME SIZE AS JOLT RAM CARD; WITH INSTRUCTIONS.

Power for CPU & Memory

microcomputer supply $44.95

THIS POWER SUPPLY KIT DELIVERS 5 VOLTS @ 5 AMPS, AS WELL AS +12V @ ½ AMP, -12V @ ½ AMP, AND AN ADJUSTABLE NEGATIVE BIAS SUPPLY. FOLDBACK CURRENT LIMITING---SAME SIZE AS JOLT POWER SUPPLY CARD--CROWBAR OVERVOLTAGE PROTECTION---1% REGULATION, 5V. A QUALITY COMPUTER SUPPLY..INCLUDES CHASSIS AND ALL HARDWARE, BUT LESS LINE CORD. FOR 117 VAC, 60 Hz. PLEASE ADD SHIPPING FOR 8 POUNDS.

91L02
Low Power
NMOS static
1K RAMS

WHILE THEY LAST!
SPECIAL!
$2.35

SEND FOR OUR FREE FLYER IT'S FILLED WITH GOODIES !!!

CAVE grafix™

10000 uF at 10V!

$1.25 THE PERFECT TONIC FOR POWER SUPPLIES WITH A FARAD DEFICIENCY. AXIAL LEAD PACKAGE



Microprocessor Based Analog

Roger Frank
1801 E Girard #247
Englewood CO 80110

An analog signal is typically a voltage level . . . which corresponds to measurement of some physical variable.

Analog signals can be processed with only a minimal addition of hardware to a system.

Analog input and output capabilities, when added to a microcomputer, can greatly expand the power of the home or hobby computer. Inherently, the microprocessor is a digital device, ideal for control of discrete (on or off) input and output levels. However, many analog signals can also be processed with only minimal additional hardware. With this addition, such devices as temperature sensors or photocells can be monitored, and output peripherals such as oscilloscopes and audio amplifiers can be added to the microprocessor.

Taking traditional approaches to analog to digital conversion can be very expensive to the hobbyist. Hundreds of dollars could be spent, but this would yield only high speed or resolution. For the amateur, typically eight bits of accuracy is sufficient, and speed is not a critical factor. The brightness of the sun, the temperature of the room, or the moisture of the front lawn do not change very rapidly. By allowing the microprocessor to do most of the work involved in the conversion, a simple, inexpensive circuit can convert an analog input to a digital word in less than a millisecond. The overall cost can be kept under $20 for four channels of analog input.

Two techniques of analog to digital conversion are easily accomplished by a microprocessor: the ramp and successive approximation methods. In each case, the task is to generate a digital word, apply it to a digital to analog converter (DAC), and compare the analog output of the DAC to the analog input to be converted. Based on the results of the comparison, the next digital word to the DAC is generated.

Traditionally, several gates, up-down counters, and clock generators are used to achieve the conversion. This approach is much more expensive than using the microprocessor to implement the same functions, using no external TTL logic in the conversion at all.

The Ramp Technique

The simplest approach is the ramp technique. It has the advantage of needing the least code in the microprocessor, but the disadvantage of being the slowest, some 15 times slower, on the average, than the successive approximation approach discussed later. For many applications, where speed is not critical, this approach may be best. Since the ramp technique is conceptually easiest to understand, it will be examined closely first.

Figure 1 shows the block diagram of the AD conversion system. Unlike hardware approaches, the identical components can be used for successive approximation, ramp, or tracking conversion algorithms. The hardware can be tailored, by software, to meet speed or accuracy requirements of the overall system.

To understand the circuit, assume in figure 1 that the analog input to the + input connector of the comparator is 2.00 V, and that all zero bits are applied to the DAC's digital inputs. The DAC's output will be 0 V at the comparator's – input connector. The comparator's output will be a 1 bit, which is applied to the microprocessor through an input port. The software, by reading and testing the input port, knows if the digital word applied to the DAC is too large or too small. In this case, the 1 bit read at the input port means "too small" and the microprocessor will increment the digital word at the input to the DAC. The output of the DAC increases by a small amount each time the comparator says "too small," until the DAC generated analog voltage just exceeds the "unknown" input voltage. At that moment, the comparator output will be read as a 0 bit, and the digital equivalent of the analog input voltage will be present at the input to the DAC.

This sequence, using an eight bit DAC, generates a ramp voltage at the input to the comparator with each step 1/256th of the

/Digital Conversion

A challenge: Write a program to send data to the DAC at regular intervals, connect the DAC output to a high fidelity amplifier, and play music with the DAC as a waveform generator.

full scale voltage. In this application, a five volt full scale is typical, so each step would be about 19.5 millivolts. Using the Motorola MC6800 microprocessor, a routine to accomplish this simple conversion would be as shown in listing 1.

Note that with the MC6800, IO is treated as a memory location, so it is simple to directly implement the algorithm. For the Intel 8008, a similar sequence could be used, as shown in listing 2. In this example, Register B will have the eight bit digital equivalent of the analog input when the sequence is complete.

The Successive Approximation Method

A faster technique, which always takes the same number of passes through the decision making loop, is the successive approximation method. The hardware is exactly the same, but instead of changing the least significant bits in incrementing fashion (19.5 millivolts per step), this method changes the most significant bits, one at a time, and very quickly homes in on the correct digital word.

Using the same example, with 2.00 V applied to the "unknown" input of the comparator, the sequence is like this. First, the most significant bit, bit 7, is set to a one in the DAC. The output of the DAC immediately goes to half scale, or 2.5 volts. (Remember that bit 7 represents 2**7 or 128 times the least significant bit's weight of 19.5 mV, which is about 2.5 volts.) Right away, the microprocessor knows that in the final digital word, bit 7 will be a zero, since the comparator is already saying "too high" with that bit only set in the DAC. The microprocessor removes bit 7 from the DAC and sets bit 6 to a one. Now the DAC output of 1.25 V is compared to the 2.00 V "unknown" input to the comparator, and the processor quickly learns that bit 6, by itself, is "too low," since 1.25 V is less than

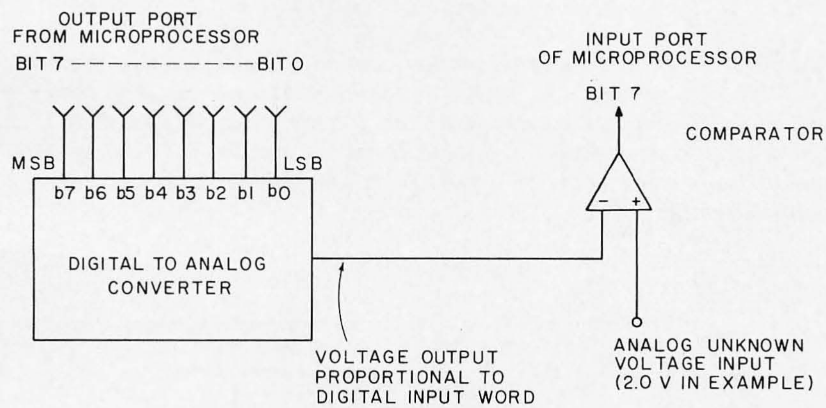


Figure 1: The microprocessor controlled analog digital conversion system consists of an 8 bit DAC output which is compared against the unknown input.

```
RAMP    CLR  DAC      start conversion at zero;
RLOOP   INC  DAC      increment output voltage;
        TST  COMP     test comparator input of bit 7 (N);
        BMI  RLOOP    back for more until done;
        RTS           return to caller;
```

Listing 1: The ramp method of conversion, specified as a symbolic assembly language program for the Motorola 6800 central processor.

```
RAMP    XOR  A        clear the accumulator with XOR;
        LBA           clear B from A;
LOOP    INC  B        increment DAC input word by one;
        LAB           move to accumulator for output;
        OUT  DAC      output to DAC device code;
        INP  COMP     input from comparator device code
        JTS  LOOP     using sign bit for comparator;
        RET           return when done;
```

Listing 2: The ramp method of conversion, specified as a symbolic assembly language program for the Intel 8008 processor.

```
1   SUCAPPRX   CLR    A         result will be in A;
2              LDAB   #$80      rotating mask, most significant first;
3   NEXTBIT    ABA              apply trial bit to A with addition;
4              STAA   DAC       send it to the output DAC latch;
5              LDAA   COMP      read the comparator output;
6              ANDA   #$80      check sign bit with comparator output;
7              BNE    RETAIN    if low then retain trial bit;
8              LDAA   DAC       recover the DAC word;
9              SBA              restore zero to last trial bit;
10             BRA    MSHIFT    then go shift the rotating mask;
11  RETAIN     LDAA   DAC       keep the trial bit as logical one;
12  MSHIFT     ROR    B         rotate the mask;
13             BCC    NEXTBIT   on eighth rotate, carry set
14             RTS                  so return from the conversion;
```

Listing 3: A successive approximation conversion, specified as a symbolic assembly language program for the Motorola 6800 processor. This program was adapted from a Motorola application note on the subject. Note that for fast processors or slow operational amplifiers (such as the 741), a delay loop should be inserted between lines 4 and 5 of this program to allow the output to settle.

```
TESTPGM   LDAA   #$00   load test value for DAC;
          STAA   DAC    and store it in the DAC;
          RTS           then return to caller;
```

Listing 4: A test program which can be used to load the immediate value of 0 into the DAC output port. The symbolic location DAC is assumed to be the output port address.

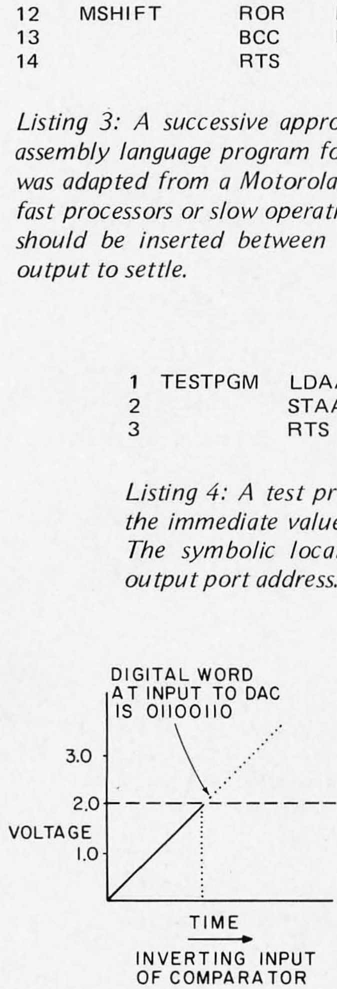


Figure 2: A ramp conversion starts at zero voltage output and increases the voltage until it equals or just exceeds the unknown input. For larger input voltages, conversion takes longer since the program must cycle through all the intermediate values from zero to the final binary word.

2.00 V. In this case, the processor leaves bit 6 on and adds the bit with lesser significance, bit 5. With bit 6 and bit 5 on, the DAC output voltage is 1.87 V, still too low. Thus, bit 5 also is left on and the next bit in line is tried.

The algorithm is this: simply try a bit, starting at the most significant. If the DAC generated voltage exceeds the "unknown," remove that bit only, else keep it. Try the next bit, repeating the process until all bits have been determined. In this case, eight passes through the loop will result in the complete digital equivalent of the unknown analog voltage input in a matter of milliseconds.

This faster technique has been implemented with the MC6800 microprocessor with the sequence shown in listing 3. A sustained rate of 1000 conversions per second has been achieved.

An actual circuit to implement these techniques is shown in figure 2. The circuit uses an inexpensive Motorola MC1408L-8 digital to analog converter, which converts digital inputs to a current output at pin 4. Current output, which is subsequently converted to a voltage, is typical with DACs.

The circuit to the left of the DAC is a simple zener diode voltage regulator. The zener maintains a constant voltage drop across the resistor R1, since the right side of the resistor is at virtual ground. The current through R1 is the reference current, which is either absorbed internally or steered out the DAC's pin 4. How much current leaves the DAC is a function of the digital input word applied on pins 5 through 12.

The current cannot be compared to the unknown analog input voltage without some conversion. Dig out your operational amplifier articles and you'll realize that the LM301 is functioning as a *current to voltage* converter, which changes the 0 to 2 mA output of the DAC into a 0 to 5 V voltage. This voltage, after a little filtering, is then applied to an LM311 comparator.

The LM311 has the useful feature of having an analog comparator input, but a TTL compatible (open collector) output. The LM311 output can be directly applied to an input port of the microprocessor for program controlled evaluation. Resistors R6 and R5 add a little hysteresis to the comparator and, like the filtering components C1, C2, C4 and R7, are recommended, though not absolutely essential to the operation of the circuit. Similarly, a 741 type opamp can be used in place of the LM301, but the circuit will take longer to convert the current output of the DAC into a stable voltage at the input to the LM311.

Circuit calibration is simple and consists of only one adjustment. First apply all zeros to the digital input to the DAC. The voltage at pin 6 of the LM301 should be very nearly zero volts. If it isn't, check your circuit carefully. If off by only a few millivolts, a small offset current could be injected into the input of the LM301 to make it exactly zero volts, but for eight bit accuracy this should not be necessary. Now apply all 1 bits to the DAC input. The output of the current to voltage converter should now be adjusted to 5.00 V with resistor R4. With this setting, you have calibrated to the 19.5 mV/b specification used in the examples.

Expansion of this circuit, once the single channel version is complete, is straightforward and very inexpensive. For example, each additional channel of analog to digital conversion can be added with only an additional comparator. Each added LM311 has its output connected to a separate input port bit, up to eight channels per port for an 8 bit processor. Then in software, choose the channel of interest by logically masking out

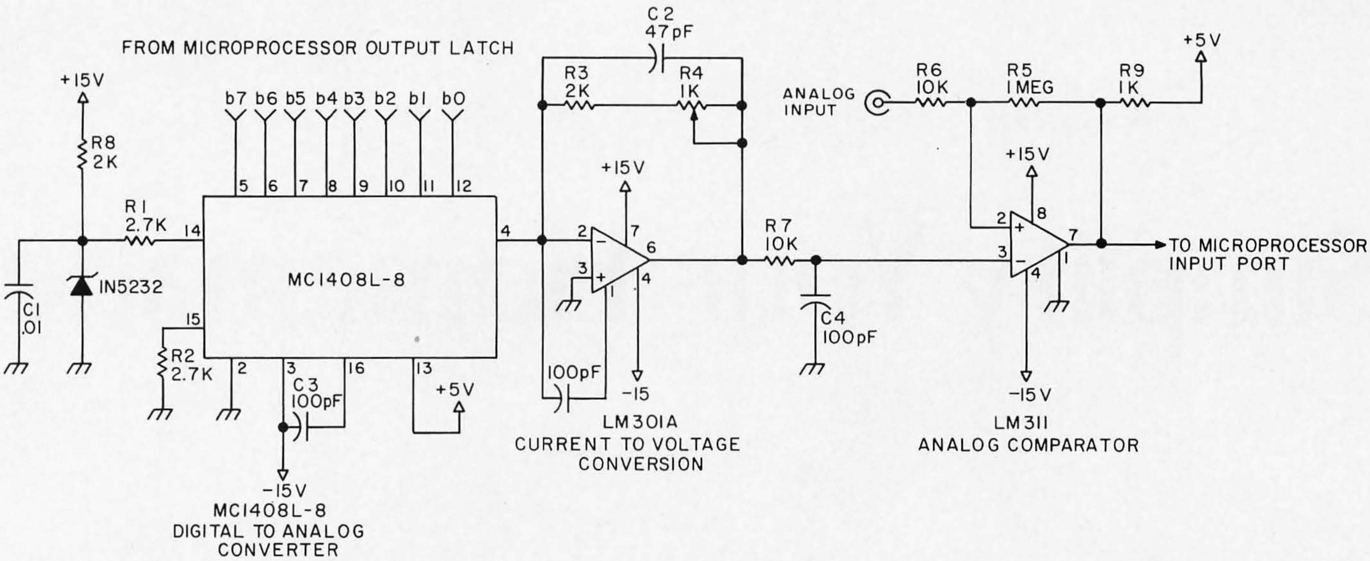


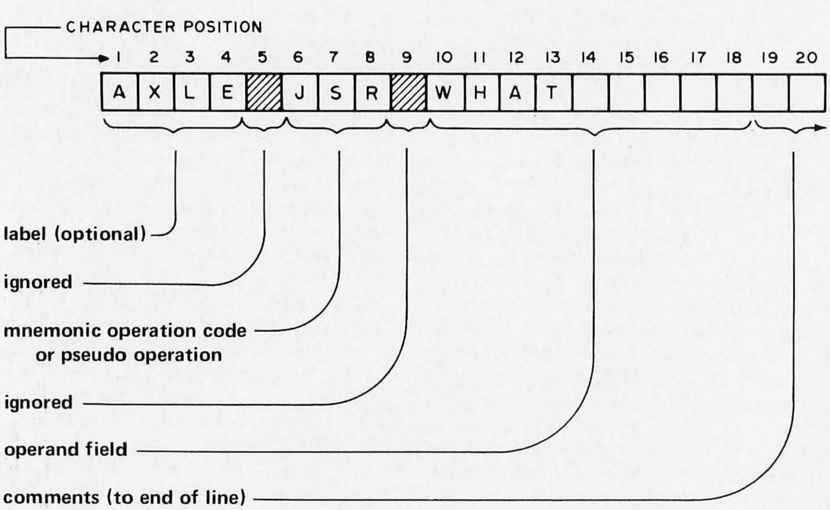
Figure 3: Schematic of the circuit used for 8 bit conversions. This hardware can be used for either the ramp or successive approximation methods described in this article.

all the other channels. Here the LM339 can be used to have four comparators, and four channels of AD, in one package. Similarly, at no charge, this circuit can be used as a source of digitally programmed analog voltage to deflect an oscilloscope trace or act as a computer controlled function generator, producing extremely complex waveforms, if desired. Another use could be a keyboard controlled power supply with suitable current gain added to the DAC output.

These techniques and this inexpensive circuit open a wide world of analog interfacing to the microprocessor hobbyist. Now the home computer can go beyond the number crunching, logic control functions and talk to the real world on its own analog terms.■

Simplify Your Homemade

Gregory C Jewell
11855 Southeast 188th
Renton WA 98055



Notes: 1. A semicolon (;) in line position 1 indicates the whole line is a comment and will be ignored by the assembler.
2. If the .AS or .AZ pseudo operations are used, the operand field can be as long as required.

Figure 1: Summary of simplified assembler source format. This figure illustrates the fixed field format. The label field is used to define symbols, the operation code field is used to specify a mnemonic operation code or a pseudo operation, and the operand field is used to contain information according to the format of figure 2. Comments may be written by starting a line with a semicolon in position one, or following the operand field with the desired comments.

Our primary goal in the design of a simple assembler is to eliminate the need to parse a line in order to determine what information is contained in that line. Rather than asking "What are you trying to give me?", our assembler will demand, "I know where I am, so give me what I want."

The assembler described here is a three-pass assembler. The first pass compiles a symbol table; the second pass outputs the generated machine code, and the third pass produces a hexadecimal listing of the generated machine code with its associated addresses and source statements.

Labels

The first step on our path to simplicity is a major one even though its impact on our program writing will be slight. We will specify that all labels should have a fixed length of four characters with a restriction that the first character should be alphabetic. Although not the main objective, requiring fixed length labels adds the feature of allowing embedded blanks in the labels. Figure 1 illustrates the fixed field format of the simplified assembler. In the example, a line of assembler input is shown in the boxes, with shaded boxes indicating blanks. The label AXLE is shown on a statement containing the JSR operation code with operand WHAT.

Six or eight characters is a popular maximum for assembly language labels; however, our four character labels will save memory space and speed up the task of searching for a label in the symbol table.

The simplified assembler will demand "I know where I am, so give me what I want."

Assembler

A label is defined when it appears for the first time in a statement of the program which is being assembled. A label is not required for every statement. However, if the first character position of the statement is found to have an alphabetic character, then the first four columns define a new label for the symbol table. If the first position is a blank, then the assembler should ignore the remaining positions of the label field. This is an example of what is called a fixed field syntax because we always expect a label or no label at all in these positions. Programming of the assembler is simplified by use of this limitation. The need for parsing has been nearly eliminated by this single requirement of fixed-length labels. But let's take a few more steps.

Operation Codes

As in commercially available assemblers, the next field on each line of the program being assembled is an operation code field. This field is separated by a blank character position from the label field, and thus begins in the sixth character position of our fixed field input format. In the operation code field, the assembler can find two types of information: an assembler pseudo operation or a mnemonic operation code for machine instructions.

Pseudo Operations

A mnemonic operation code is a symbol which the assembler in most instances will translate into a machine instruction. A pseudo operation code is a similar symbol which looks very much like a mnemonic operation code. However, the pseudo operation does not normally generate machine instructions and is used instead to control how the assembler will generate code. All assemblers have pseudo operations. Ours is no exception. When choosing pseudo operations, the goal of simplicity should be kept in mind. Most likely we will be able to get by without many of the fancy or powerful pseudo operations that add bulk and complexity to the assembler program.

I have defined nine basic pseudo operations for my assembler. All begin with a period so that the assembler program need only examine the first character to determine if the mnemonic is a pseudo operation. This speeds address calculation during the first pass since all other PACE instructions generate a single word of code. It also aids

Mnemonic	Description	Action
.SA	starting address	Defines the address for the next instruction. The assembler must know where to start assigning code whether by default or instruction. Similar to the ORG pseudo-op of other languages.
.RS	reserve storage	Saves space for specified number of words.
.XW	hexadecimal word	Loads specified hexadecimal value into location.
.AS	ASCII string	Breaks down a character string into its ASCII equivalent.
.AZ	ASCII string with zero	Same as .AS except that the ASCII code is terminated by a zero byte.
.DF	define address	Loads address of specified label into location.
.IL	inhibit listing	Inhibits listing during third pass.
.EL	enable listing	Enables listing during third pass (default condition).
.ND	end	End of source program.

Table 1: Pseudo operations.

human recognition. The nine pseudo operations are briefly described in table 1.

We now have all the information required to complete the first (address allocation) pass. It is possible to identify a label and calculate its address, since PACE has fixed length instructions. The label and its associated address are stored in the symbol table sequentially. A symbol definition requires three words, since we must store two words for the name and one word for the address. If desired, at the end of the first pass the labels may be sorted by the first character (it's surprising how close this

Table 2: An example of the output of an assembler implemented according to this definition. This assembly shows a memory test program written for the author's system. Bearing in mind all the limitations placed upon the source format to simplify writing the assembler, note that the listing looks like a "typical" output of an assembler. Note the frequent use of comment lines (starting with a semicolon) to explain various aspects of the program. The program uses the author's 3 character mnemonics instead of the PACE mnemonics, and the pseudo operations are shown in table 1.

```
END FIRST PASS,        0 ERRORS DETECTED

SYMBOL TABLE

 DATA     000E
 DSPL     0024
 ERR      0020
 LIM      0025
 NEXT     001D
 RITE     0001
 READ     0006
 REED     0015

END OF SYMBOL TABLE,     8 LABELS

                  ; TWO-PART MEMORY TEST
                  ; (1) ADDRESS-DATA CHECK
                  ;     WRITE A UNIQUE NUMBER IN ALL LOCATIONS
                  ;     IF A USED ADDRESS LINE IS BAD, THEN AT LEAST
                  ;       ONE ERROR WILL OCCUR
                  ;
                       .SA 0
0000 5226              LIM R2,26      LOAD STARTING ADDRESS OF TEST
0001 DA00 RITE         STA R2,(R2)    WRITE ADDRESS INTO LOCATION
0002 F922              SNE R2,LIM     MEMORY LIMIT REACHED?
0003 1902              JMP READ       YES
0004 7A01              AIS R2,1       NO. INC INDEX
0005 19FB              JMP RITE
                  ;
                  ; READ BACK UNIQUE NUMBERS
                  ;
0006 5226 READ         LIM R2,26      RELOAD STARTING ADDRESS
0007 FA00              SNE R2,(R2)    COMPARE, SKIP IF ERROR
0008 1901              JMP .+2
0009 1916              JMP ERR
000A F91A              SNE R2,LIM     MEMORY LIMIT REACHED?
000B 1902              JMP DATA       YES, GO TO NEXT PART OF TEST
000C 7A01              AIS R2,1       NO. INC INDEX
000D 19F9              JMP READ+1
                  ;
                  ; (2) SHIFT-ONE DATA CHECK
                  ;     TEST WORD HAS A SINGLE BIT SET
                  ;     WRITE TEST WORD IN ALL LOCATIONS
                  ;     TEST ALL BIT POSITIONS
                  ;
000E 5001 DATA         LIM R0,1       INITIALIZE TEST WORD
000F 5226              LIM R2,26      LOAD STARTING ADDRESS
0010 D200              STA R0,(R2)    WRITE TEST WORD
0011 F913              SNE R2,LIM     MEMORY LIMIT REACHED?
0012 1902              JMP REED       YES
0013 7A01              AIS R2,1       NO. INC INDEX
0014 19FB              JMP DATA+2
                  ;
                  ; READ BACK TEST WORD
                  ;
0015 5226 REED         LIM R2,26      RELOAD STARTING ADDRESS
0016 F200              SNE R0,(R2)    COMPARE, SKIP IF ERROR
0017 1901              JMP  +2
0018 1907              JMP ERR
0019 F90B              SNE R2,LIM     MEMORY LIMIT REACHED?
001A 1902              JMP NEXT       YES
001B 7A01              AIS R2,1       NO. INC INDEX
001C 19F9              JMP REED+1
001D 2802 NEXT         SHL R0,1       SHIFT TEST WORD
001E 45F0              BOC 5,DATA+1   WRITE NEW TEST WORD IF NONZERO
001F 5E00              CPY R2,R0      TEST COMPLETE, DISPLAY 0 ERRORS
                  ;
                  ; ERROR ROUTINE: DISPLAY BAD LOCATION
                  ;
0020 5C80 ERR          CPY R0,R2
0021 B102              STI DSPL       LOAD DISPLAY REGISTER
0022 D903              STA R2,LIM +1  SAVE TEST DATA FOR REFERENCE
0023 0000              HLT
0024 8009 DSPL         .XW 8009       ADDRESS OF DISPLAY REGISTER
0025 03FF LIM          .XW 3FF        MEMORY LIMIT = 1K
                       .ND

END THIRD PASS,        0 ERRORS DETECTED
```

comes to actually alphabetizing the labels) and listed with their addresses. The sample assembly of table 2 shows the result of such a sort.

Mnemonic Operation Codes

The next step toward simplification is to specify that all mnemonic operation codes should also have a fixed length. National Semiconductor Corporation, PACE's manufacturer, suggests mnemonics containing from two to five characters. Even if we use the manufacturer's suggested mnemonics and specify a fixed length of five characters, the indirect notation @ would probably throw a wrench into the works since the @ usually directly precedes the label rather than immediately following the mnemonic.

I chose to define a set of 3 character mnemonics. This saves memory space and speeds up the search for mnemonics in the table of operation codes. The three characters of the mnemonic operation code can be stored in one and a half words (3 bytes) and the binary opcode may be kept in the remaining byte. There is nothing magic about mnemonics; they are simply aids to remembering the instructions. It's your computer, so you might as well use your own mnemonics – unless you plan to make your assembler commercially available. Table 3 shows the correlation between the

An effective address is a combination of an addressing mode and a displacement.

Table 3: Correlation between manufacturer's suggested mnemonics and the author's 3 character mnemonics.

	Manfacturer's Suggested Mnemonics	Description	Author's Mnemonics
1.	JMP	jump	JMP
2.	JMP@	jump indirect	JMI
3.	JSR	jump to subroutine	JSR
4.	JSR@	jump to subroutine indirect	JSI
5.	SKG	skip if greater	SGT
6.	SKAZ	skip if AND is zero	SAZ
7.	ISZ	increment and skip if zero	ISZ
8.	DSZ	decrement and skip if zero	DSZ
9.	LD@	load indirect	LDI
10.	ST@	store indirect	STI
11.	LSEX	load with sign extended	LSX
12.	AND	logical AND	AND
13.	OR	logical OR	IOR
14.	SUBB	subtract with borrow	SBB
15.	DECA	decimal add	DCA
16.	AISZ	add immediate, skip if zero	AIS
17.	LI	load immediate	LIM
18.	XCHRS	exchange register and stack	XRS
19.	CFR	copy flags into register	CFR
20.	CRF	copy register into flags	CRF
21.	PUSH	push register onto stack	PSH
22.	PULL	pull register from stack	PUL
23.	CAI	complement and add immediate	CAI
24.	SKNE	skip if not equal	SNE
25.	LD	load	LDA
26.	ST	store	STA
27.	ADD	add	ADD
28.	RXCH	register exchange	RGX
29.	RCPY	register copy	CPY
30.	RADD	register add	RAD
31.	RADC	register add with carry	RAC
32.	RAND	register logical AND	RND
33.	RXOR	register exclusive-OR	XOR
34.	BOC	branch on condition	BOC
35.	RTS	return from subroutine	RTS
36.	RTI	return from interrupt	RTI
37.	PUSHF	push flags onto stack	PSF
38.	PULLF	pull stack into flags	PLF
39.	HALT	halt	HLT
40.	SFLG	set flag	SET
41.	PFLG	pulse flag	PLS
42.	SHL	shift left	SHL
43.	SHR	shift right	SHR
44.	ROL	rotate left	ROL
45.	ROR	rotate right	ROR

All assemblers have pseudo operations. This one is no exception.

manufacturer's suggested mnemonics and the 3 character mnemonics which I selected to simplify my assembler.

Instruction Groups

So far we have defined a 4 character label field and a 3 character mnemonic field. To make the program readable, we'll allow a single character (blank) after each field and a semicolon in the first character position (column one) to signal a comment line. Our assembler now expects either a blank, a semicolon, or an alphabetic character in the first position. As noted earlier, if the first position of a line contains an alphabetic character, then a label exists in the first four positions. The fifth position is ignored. The sixth through eighth positions contain the operation code or pseudo operation mnemonic and the ninth position is ignored. What does the assembler expect in the tenth position? To answer this question, we must collect instructions with similar binary and source formats into instruction groups. The only variation within an instruction group is the mnemonic operation code. Figure 2 lists the ten PACE instruction groups.

After the instruction group is determined, our assembler will know exactly what to look for and where to find it. For example, if the instruction is in group three, the tenth character position is ignored (allowing you to specify RO, AO, XO, or whatever pleases you at the time), a digit less than four is expected in the eleventh position; the twelfth position is ignored, and the destination (DEST) field begins in the thirteenth position. If the instruction is in group four, then the assembler expects to find a digit less than four in the eleventh and fourteenth positions. If the instruction is in group seven, then the assembler's worries are over, since such instructions have no operands.

Destination Field

The destination field (DEST) is required to determine the effective address. An effective address is the combination of an addressing mode and a displacement. The four PACE addressing modes are program counter relative, relative to register R2 used as an index, relative to register R3 used as an index, and base page. All addressing modes of the destination field entries (destination modes) listed in table 4 are program counter relative except the last two: (R) is index mode and *K is base page mode. The index and base page modes are limited primarily by my own biases and could be chosen differently in your own version of such an assembler. As with all other fields of a personal assembler, the DEST field should be tailored to your own preferences. The modes of table 4 are sufficient while maintaining the goal of simplicity.

Figure 2: PACE Instruction groups.

Group	Instructions	Binary Format (bits 15 14 13 12 11 10 09 08 07 06 05 04 03 02 01 00)	Operand Format (Position 10)
0	JMP,JMI,JSR,JSI,SGT,SAZ,ISZ,DSZ, LDI,STI,LSX,AND,IOR,SBB,DCA	OP | XR | DISP	DEST*
1	AIS,LIM,CAI	OP | R | IMMEDIATE	R,K
2	XRS,CFR,CRF,PSH,PUL	OP | R | NOT USED	R
3	SNE,LDA,STA,ADD	OP | R | XR | DISP	R, DEST*
4	RGX,CPY,RAD,RAC,RND,XOR	OP | DR | SR | NOT USED	R,R
5	BOC	OP | CC | DISP	M,DEST*
6	RTS,RTI	OP | DISP	K
7	PSF,PLF,HLT	OP | NOT USED	none
8	SET,PLS	OP | FC | P | NOT USED	M
9	SHL,SHR,ROL,ROR	OP | R | N | L	R,K or R,K,L

R = R0, R1, R2 or R3
0 ⩽ M ⩽ F
0 ⩽ K ⩽ FF
L = "L" (letter L)

*** See Modes of the destination field, table 4.**

Again, by examining only the first character of the field, the assembler can determine if the DEST field has a label, a specified displacement, an index register, or a base page value. The + or – extension after the label will always be in the same relative position since we have declared that all labels contain four characters. If the first character of the DEST field is an alphabetic character, then the first four characters of the field form the label; and, if there is an extension, the + or – will always be the fifth character of the field.

Example

Table 2 shows the output of the first and third passes of a memory test program. It looks general even though strict rules were applied. The execution time is approximately 1.5 seconds for each 1 K of memory tested. Notice the destination LIM +1 in statement line 57. LIM+1 would have produced an UNDEFINED LABEL error. The trailing blank is part of the label.

If you desire simplicity and can live with LIM +1 rather than LIM+1 then you might implement the rules I have presented (or your own variation) in your homemade assembler.

Conclusion

The simplified homemade assembler's source language is now completely defined in a way which is simple and easy to implement, yet probably adequate for all our programming needs. Except for the .AS and .AZ pseudo operations, we have eliminated the need for parsing, mainly by specifying a fixed label length (with embedded blanks) and a fixed mnemonic length. Other simplifications were achieved by selecting only basic pseudo operations and destination modes. By using these techniques, you should have your homemade assembler running by tomorrow.■

Table 4: Modes of the destination field (DEST).

DEST		Description
LABEL		symbolic
LABEL+K		symbol relative
LABEL–K		symbol relative
.+K	(here plus K)	program counter relative
.–K	(here minus K)	program counter relative
(R)		index register
*K		base page

0 < = K < = FF
R = R2 or R3

REFERENCE

PACE Technical Description
National Semiconductor Corp
Santa Clara CA 95051
Publication number 4200078A
June 1975

GLOSSARY

ASCII: American Standard Code for Information Interchange. A 7 bit code used by many machines.

Assembler: An assembler is a program which accepts a symbolic representation of some computer program and transforms it into one which can be executed by a computer. The symbolic representation is called a source program; the executable representation is called an object program.

Character Position: Each line of the source program which is read by the assembler is a character string. In a fixed field syntax, the character positions are numbered (in this case, from 1 to the end of the line). Each field of the format is a group of characters specified by number, such as the label field which is positions 1 to 4 of a line in the example of this article.

Mnemonic: A technique to assist human memory. A mnemonic term is an abbreviation or acronym used instead of numeric codes in order to facilitate easy recognition. Example: BOC for Branch On Condition rather than 4.

Parsing: The breaking down of a general character string into its structural forms. This requires syntax rules for the computer language analogous to the grammar rules for English that define "subject," "predicate," "object," and so forth. In this assembler, we simplify syntax rules by requiring fixed positions for each piece of information on a line which eliminates the need for parsing.

Pass: An assembler typically must look at the entire data of a program several times. Each pass of an assembler is one complete scan through the program data. In the simplest home brew assemblers using audio cassette mass storage, each pass will require manual intervention to rewind and restart the appropriate tape cassette drive.

Pseudo operation: A group of characters having the same general form as a computer instruction, but never executed by the computer as an actual instruction. Pseudo operations are instructions to the assembler.

Source Program: A program coded as a human readable character string in some programming language, which must be translated into machine language.

Symbol Table: A dictionary relating one set of symbols to another set of symbols or numbers. The assembler builds a table of labels used in the assembly language program and assigns memory locations (addresses) to those labels.

Continued from page 16

unless the convention of using a slashed zero is used as well. To avoid all this perhaps it would be best if the standard symbol for ohms, the Greek letter Omega (Ω) is used, or that the word "ohm" be spelled out to avoid confusion.

Michael S Maiten
Los Angeles CA

You have a point, which was also made by several others. In future BYTEs you can expect to see the term "ohm" used where resistance values are specified.

SCIENTIFIC APPLICATIONS, ANYONE?

I would like to suggest that the various scientific applications of the various microcomputer systems available should receive some time in the Foreground. Floating point processing, implementation of hardware and software approaches to trigonometric and transcendental functions, digital filtering, AD and DA conversion are all areas of application that expand the usefulness of the microcomputer into practical, real world science and engineering. It's been said before, but the applications are limited only by the user's imagination. Some simple examples might be the star locator routine used by an amateur astronomer, or the antenna simulator used by the ham radio operator for the computer-aided-design of a new antenna, or the engineer studying the response of a bandpass filter for a biofeedback system.

Truly, the possibilities are awesome.

B Humprey Jr
FPO Seattle WA

NO DeBYTEchery . . .

My real reason for writing is the BOMB . . . I know that I feel about BYTEs like the other dudes and dudesses feel about their BYTEs and man there is no way that I'm going to tear, bend, spindle or multilate my mag so that someone can get $50 bucks (to me the mag is worth $50 bucks). The point is this. . . put BOMB on a card – I'd even pay postage – but do not ask me to RAPE my BYTEs!

Do not ask me to RAPE my BYTEs!

Michael B Gamble
APO New York NY

Sorry, Michael. The BOMB form and questionnaire will remain on tear sheets for the time being. However, you (or anyone else) are free to photocopy the BOMB page by whatever means available . . . we place a notice to that effect on the pages in question. You will note, however, that we now have an easily removable reader's service "bingo" card, and a subscription service card which avoids the need to cut up BYTE for those important functions, yet preserves a magazine free of excessive insertions which interfere with reading and enjoyment of the copy.

THE ABSURDITY OF ALL POWERFUL COMPUTERS

"Could A Computer Take Over?" misses an important point: Computers now or in the next generation could never be even a fraction as reliable or intelligent as a human being. Estimates of the information capacity of the human brain range from 10**40 to 10**70 bits from what I know. The biggest computer systems today have on the order of 10**10 bits. On the basis of memory capacity only, that makes the biggest computer systems today at least 10**30 times as stupid as human beings.

Computers now or in the next generation could never be even a fraction as reliable or intelligent as a human being.

Moreover, if a computer with the capacity of the human mind did exist, its intelligence would never be realized because:

1. At error levels even millions of times smaller than the best computer today and with total redundancy of operation, the system would crash into a twitching mass of bug ridden pulses every time its vast resources are tapped.
2. Software written for it could never be debugged and proven out completely even if the hardware could be relied upon.

Paul Carrick
San Jose CA

Beware of proofs that "it can't be done." At least one eminent professor "proved" prior to 1903 that a heavier than air machine could not fly. However, more than likely any computer oriented dictatorship would instead be built upon a layer of "Cybercrud," distortions and mystical half truths about computers (see Theodor Nelson's book, Computer Lib/Dream Machines*). Part of the goal of popularizing computer technology is to help avoid such a dire turn of events.*

How close are we to a computer that can drive a car (without a crash)?

OGHAM AND STONEHENGE

It appears that Thomas C McIntire, author of "How to Save the Bytes," has re-invented Ogham, an archaic Irish character set which was in use 1500 to 1000 years ago.

Is it possible that the Irish invented Ogham for the same reason Mr McIntire did, to achieve communication with a computer? It is well known that analog computers existed in ancient times. Perhaps these

computers possessed digital capability which has gone unrecognized. I am, of course, referring to Stonehenge, an early CPU composed of monolithic chips. Similar woodhenges also existed.

Have menhirs inscribed with Ogham ever been found?

Has a menhir reader ever been discovered at Stonehenge?

H A Jones

SOME SUGGESTIONS

I would like to see BYTE be a critical eye over the amateur computing field. New machines, chips, software should be reviewed in some standardized fashion, as in the *Popular Science* new car tests. The conveniences of software availability should be gone into; Bob Baker's PDP-11/LSI story only had one paragraph on that subject. The article did not compare the quality of the instruction set to the other available micros (I'm not familiar enough with any chips other than the 8080 to make a judgment, just given the instruction set). How does its adaptability by home tinkerers compare with other machines? Don't get me wrong. I liked the article on the LSI-11, but I think Bob should have been more critical as well as doing the good job he did on explaining what's inside it. These things, while a lot cheaper than in the past, are still a major investment. I'd like to know what I'm giving up in order to get what I'm getting.

I would also like to see software articles, maybe programming style and structured programming, explained, and algorithms for the interesting or the useful. Full programs are nice but I'd also like an emphasis on "how to," ie, "take this technique and run" rather than canned routines.

Is there any such thing as a home built printer, plotter, disk drive and other exotic equipment which gets around the high-technology manufacturing by using labor intensive techniques at home? I'd like to know about it.

As spice, maybe some articles occasionally on exotic uses of computers in general – send a reporter to the ACM meetings and tell us what's happening; for instance, how close are we to a computer that can drive a car (without a crash)?

Philip Robare
Champaign IL

BYTE often gets letters which are full of suggestions about articles for the magazine. Philip Robare's is one of the best in that category. Potential authors would do well to look at some of his suggestions for articles and short subjects.

ATTENTION All Ye Alice Freaks and Other Lovers of Logical Systems. Here is documented evidence that Lewis Carroll would have read BYTE had he lived in 1976:

JABBERWOCKY.

'Twas brillig, and the slithy toves
  Did gyre and gimble in the wabe;
All mimsy were the borogoves,
  And the mome raths outgrabe.

"Beware the Jabberwock, my son!
  The jaws that BYTE, the claws that catch!
Beware the Jubjub bird, and shun
  The frumious Bandersnatch!"

He took his vorpal sword in hand:
  Long time the manxome foe he sought–
So rested he by the Tumtum tree,
  And stood awhile in thought.

And as in uffish thought he stood,
  The Jabberwock, with eyes of flame,
Came whiffling through the tulgey wood,
  And burbled as it came!

One, two! One, two! And through and through
  The vorpal blade went snicker-snack!
He left it dead, and with its head
  He went galumphing back.

"And hast thou slain the Jabberwock?
  Come to my arms, my beamish boy!
O frabjous day! Callooh! Callay!"
  He chortled in his joy.

'Twas brillig, and the slithy toves
  Did gyre and gimble in the wabe;
All mimsy were the borogoves,
  And the mome raths outgrabe.

Don't miss out on all the fun and high quality information which is found in every issue of BYTE. Subscribe today. Join the tea party and have a BYTE to eat.



If the subscription card is missing from your issue, use the coupons below.

Name ....
Address ....
City .... State............ Zip.......
☐ BILL ME ☐ Check for $12 enclosed
☐ Bill BankAmericard or MasterCharge # ....

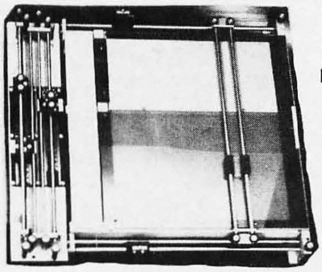
Name ....
Address ....
City .... State............ Zip.......
☐ BILL ME ☐ Check for $12 enclosed
☐ Bill BankAmericard or MasterCharge # ....

• BYTE • 70 Main St • Peterborough NH 03458

Allow six weeks for processing.

DELTA ELECTRONICS CO

POST OFFICE BOX 2, AMESBURY, MASS. 01913 Phone (617) 388-4705



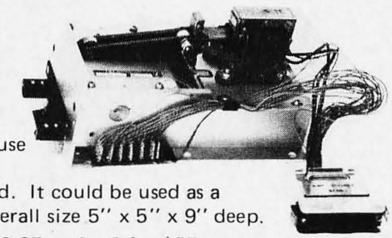
Bolt, Baranek & Newman DATA CODER

DELTA specializes in the unusual in electronics surplus, and we have once again found a highly desirable item. This is a DATA CODER (digitizer) made by Bolt, Baranek & Newman. It was made for use in the medical electronics field, but finds use in many different applications. It is used to digitize any type of data. The data (charts, maps, waveforms, drawings, game grids, etc.) is placed in the bed, and the sight is moved along the data. This generates vertical & horizontal displacement codes, which can be stored and processed by your calculator or micro-computer.

The bed size is 12¾" x 10 5/8", and has a 7" x 10½" portion which is translucent, for use as a light table. Each axis is divided into 128 increments, for a total of 16,384 discrete bits over the bed. Horiz. increment size = .10", vert. increment size = .083". Each axis generates an 8 bit code by means of wipers on an encoding board. This makes it very easy to interface to any logic family.

This item is a must for your micro-computer, as it is now possible to digitize **any** type of data. Limited quantity. Overall size is 18¾" x 17¼" x 3" high. Shipping weight 10 lbs. Comes with complete coding data. **STOCK NO. B5352 $79.95 each**

SECURITY SYSTEM CARD READER



Another unusual item from DELTA ! These card readers were made by AMP, Inc. for use in **security systems.** A stiff 2 1/8" x 3 3/8" card (credit card size) is inserted, which closes a micro-switch. A 115v AC solenoid is then energized, which pulls down a set of wipers to read through holes in the card. The wipers are arranged in 3 8 bit bytes, plus 1 bit, for 25 bits total. By using both sides of the card, 48 bits are available. Ideal for security systems---entry can be controlled by use use of a card, with an almost infinite number of combinations, rather than an easily duplicated key. As another example, an entire Social Security number, plus an entry code, could be read from a card. It could be used as a cheap bootstrap loader for micro-computers, or as a simple data entry device. Many other uses. Overall size 5" x 5" x 9" deep.

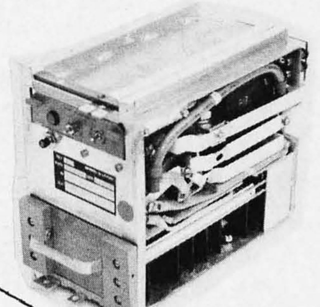
Shipping weight 6 lbs. **STOCK NO. B5353** **$19.95 each, 2 for $35**



MICROSWITCH KEYBOARD

Price slashed to $20 ! Was $45 in catalog 15, special this month at $20. Made by MICROSWITCH (Honeywell). Brand **new**, in factory cartons, some even have the top mounting plate attached. 54 keys + space bar, alpha-numeric & computer control keys. The coding is EBCDID (not ASCII) 7 level output TTL compatable, Easily converted to ASCII or other codes. The switches are mounted on one PC board & connected to a 2nd encoder board by ribbon cable. 13½" wide x 5¼", shipping weight 8 lbs. At this price, how can you pass it up? **STOCK NO. B5199** **reduced to $20 each**

5 volt, 60 amp REGULATOR



A super heavy duty 5 volt regulator made by **SPERRY/UNIVAC.** 21v DC to 35v DC input (30v nominal).Output adjustable from 4.75v to 5.25v, up to 60 AMPS ! Overcurrent protected at 65 A, overvoltage crowbar at 7.0v. Typical ripple only 50mv p-p at full load. 5¼" wide, 8½" high, 10" deep. Shipping weight 24 lbs. We only have a few, so place your order now.

STOCK NO. B5391 **$35 each**



KEYBOARD KIT

This unusual keyboard kit is made by **Micro-Switch.** It has a set of switches & space bar in a modular frame, plus 42 molded double shot keys packed in a foam carton. The keys are red, white or blue, with 8 control keys in addition to letters, numbers and many symbols. The switches are arranged in 4 rows but are easily removed or moved to other positions. This makes for a very versatile keyboard, as any number & type of keys can be arranged in any pattern to suit your individual needs. Any type of encoding can be wired. Finished size is 9¾" x 3¾" x 2" high.

STOCK NO. B6015 **Shipping weight 3 lbs.** **$19.95 each, 2 for $35**



Shipping weight 27 lbs.

Heavy Duty Power Supplies

These heavy duty power supplies are ideal as a general purpose lab supply, micro-computer supply, etc. All have a constant voltage transformer and large 18,000 mfd. 35v electrolytic filtering caps. In addition, the + and – 10v and –3.6v outputs each have seperate semi-conductor regulators, which are voltage adjustable. Each is 22¼" long, 6½" wide, 6" high.

B5313....Outputs at –30v @ 1.5 amp, +30v @ .5 amp, –10v @ 1.5 amp, +10v @ .5 amp, and –3.6v @ 1.5 amp. 85 watts total. $24.50 each

B5314....Outputs at –30v @ 1 amp, +30v @ .5 amp, –10v @ 1 amp, and +10v @ .5 amp. 60 watts total. $19.50 each

B5342....Outputs at –30v @ 1.5 amp, +30v @ .5 amp, –10v @ 1.5 amp, and +10v @ .5 amp. 80 watts total. $22.50 each

IBM 1620 COMPUTER

Our 1620 is for sale ! Complete IBM 1620 computer system with 1620 20K CPU (with all options), 2.5 megabyte 1311 DISK drive, and 1622 Card Reader-Punch. Complete with software (Fortran, SOS, scientific packages, and DOS), complete set of hardware diagnostics, logic diagrams, and manuals. Also a spare set of logic cards. The system is currently running and can be seen; however, the core memory box has at least one bad matrix line and there are several areas of memory which cannot be used unless repaired. (You could program around those areas.) System sold as-is for $3400. Phone for more information.

Send for our latest free catalog. We welcome Mastercharge & BankAmeriCard orders; we must have ALL the numbers on the card for processing. Please include sufficient postage (2 lbs. min.)—excess will be refunded. Minimum order $5.

Puzzle Time

Using the letters of the alphabet from A to Y, place a letter in each box of the 5 by 5 grid so that the sum of the ASCII codes for the letters in each column and each row is equal. To help you, the following table gives the *OCTAL* ASCII code for each letter. All codes are 7 bit ASCII.

A	= 101
B	= 102
C	= 103
D	= 104
E	= 105
F	= 106
G	= 107
H	= 110
I	= 111
J	= 112
K	= 113
L	= 114
M	= 115
N	= 116
O	= 117
P	= 120
Q	= 121
R	= 122
S	= 123
T	= 124
U	= 125
V	= 126
W	= 127
X	= 130
Y	= 131
(Z	= 132)

Answer in June issue

Robert Baker
34 White Pine Dr
Littleton MA 01460

Answer to **Space Ace** April page 12

N	F	S	U	B	R	O	U	T	I	N	E	S	U	A	P
O	R	E	C	O	R	D	I	M	E	N	S	I	O	N	R
T	O	U	T	P	U	T	R	O	T	A	R	E	P	O	O
W	R	I	T	E	X	T	E	R	N	A	L	R	U	I	G
E	D	O	C	E	D	O	C	N	E	O	I	E	N	S	R
E	L	B	A	I	R	A	V	S	G	U	B	W	C	I	A
C	A	I	O	D	N	E	P	I	N	T	R	I	H	C	M
N	T	N	M	R	T	A	C	O	A	A	A	N	E	E	H
E	C	T	A	P	C	A	I	C	M	G	R	D	N	R	T
L	O	E	T	E	L	S	R	A	E	O	Y	N	T	P	I
A	M	G	A	A	S	I	N	L	L	T	N	I	R	P	R
V	P	E	D	E	D	P	C	L	I	O	L	F	Y	A	A
I	L	R	R	L	O	A	O	I	S	L	A	E	R	R	G
U	E	P	E	T	Y	P	E	O	T	I	N	U	B	R	O
Q	X	I	S	C	A	L	A	R	L	I	T	E	R	A	L
E	F	U	N	C	T	I	O	N	G	I	S	S	A	Y	L

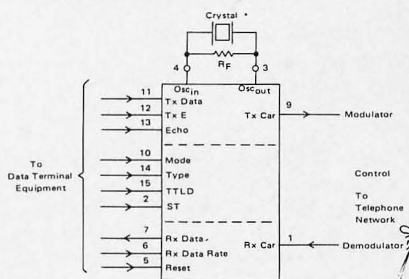
MC14412 UNIVERSAL MODEM CHIP

MC14412 contains a complete FSK modulator and de-modulator compatible with foreign and USA communications. (0-600 BPS)

FEATURES:
- .On chip crystal oscillator
- .Echo suppressor disable tone generator
- .Originate and answer modes
- .Simplex, half-duplex, and full duplex operation
- .On chip sine wave
- .Modem self test mode
- .Selectable data rates: 0-200, 0-300, 0-600
- .Single supply
  VDD=4.75 to 15VDC - FL suffix
  VDD=4.75 to 6 VDC - VL suffix

TYPICAL APPLICATIONS:
- .Stand alone - low speed modems
- .Built - in low speed modems
- .Remote terminals, accoustic couplers

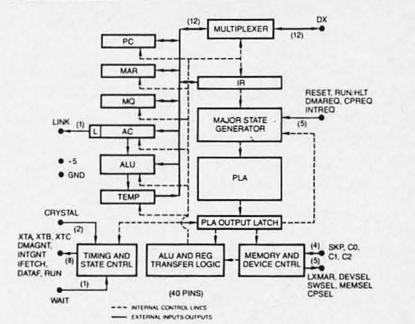
MC14412FL................................ $28.99
MC14412VL................................ $21.74
6 pages of data........................ .60
Crystal for the above......$4.95





MC14411 BIT RATE GENERATOR.

Single chip for generating selectable frequencies for equipment in data communications such as TTY, printers, CRT s or microprocessors. Generates 14 different standard bit rates which are multiplied under external control to 1X, 8X, 16X or 64X initial value. Operates from single +5 volt supply. MC14411....................... $11.98
4 pages of data................................ .40
Crystal for the above.......................... $4.95



IM6100 CPU. Intersils' 12 bit CMOS CPU chip is the microprocessor which recognizes the famous PDP8/E instruction set. Single power supply, 4-7V@ 400uA. $52.50
Full data packet................................$4.00

TELETYPE CODE CONVERSION CHIP

MM5220BL converts 5 level Baudot into 8 level ASCII. Use this chip to make your old TTY talk to your new computer. MM5220BL.................................. $18.00
Specs for the above...................... .30

MOS TIME BASE KIT.

Only 1" X 1.5". Input 5 to 15 VDC, output is 60HZ square wave for portable or mobile clocks. PC board is drilled! MTBK-60HZ....................... $6.88

78H05 Voltage regulator. Fairchild 5V, 5A, TO-3 regulator. Take care of those heavy current requirments without separate regulator/pass transistor combinations. Use it with the same ease of instalation as the 309K(same pin arrangement.).........with specs............... $11.25



LM317 Voltage Regulator. 1.5A, 3 terminal adjustable regulator in TO-3 case. Adjusts from +1.2V to +37V. Complete overload protection. .1% load regulation, .01%/V line regulation. No need to stock assorted regulators - just stock resistors.....................$4.99
Specs for the above.................. .70

DATA BOOKS BY NATIONAL SEMICONDUCTOR

DIGITAL. Covers TTL, DTL, Tri-State, etc. $3.95
LINEAR. Covers amplifiers, pre-amps, op-amps, .. $3.95
LINEAR APPLICATIONS. Dozens of application notes and technical briefs covering the use of op-amps, regulators, phase locked loops and audio amps............... $3.25
CMOS Gates, Flip Flops, registers, functional blocks $3
VOLTAGE REGULATORS. A must for anyone making a power supply. Complete theory including transformers, filters, heat sinks, regulators, etc................. $3.00
MEMORY. Information on MOS and Bipolar memories: RAMS, ROMS, PROMS and decoders/encoders.. ... $3.95
INTERFACE. Covers peripheral drivers, level translators, line driver/receivers, memory and clock drivers, sense amps display driver and opto-couplers.................. $3.95
(Outside U.S., add postage for 1.5lbs)

DATA BOOKS FROM FAIRCHILD.

uA Linear. 776 pages of data and applications for Fairchild linear ICs. Great value..................$4.25
MOS/CMOS/nMOS/pMOS/CCD. Data and applications on MOS and charge coupled devices including preliminary data on new and future offerings. Want to know about 16K charge coupled line addressable memories?.........$3.95

AMP L'ANNY Says

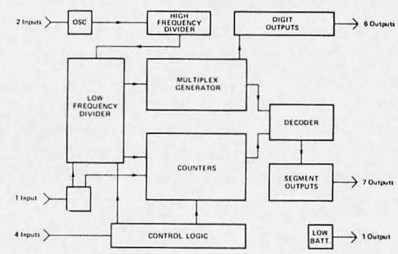
UP A TREE 'CAUSE YOU CAN'T FIND THOSE NEW DEVICES YOU'VE BEEN READING ABOUT? ....TRY TRI-TEK!!!..............

IM6508 CPE CMOS RAM

1K X 1 Bit in 16 pin DIP offers micro watt operation for on-card battery back up operation. This keeps contents of memory from disappearing on power down. On-chip address registers improve system performance and reduce package count. IM6508CPE....................$7.75

IM6518 CPN CMOS RAM.

1K X 1 Bit like the 6508 except two additional chip select pins have been added for reduced interconnects on larger memory layouts. 18 pin DIP package.
IM6518CPN.................................. $7.75
6 pages of data for above......... .90



ICM7205 IPG CMOS STOPWATCH CHIP

24 pin DIP package has on-chip oscillator and 6 digit direct drive capability. All you add is crystal, trimmer, battery, switches and LEDs. Has START/STOP, TAYLOR or SPLIT modes. 59 min, 59.99 sec range. Low battery indicator. ICM7205 IPG..................... $19.95
6 pages of data........... .90

PRECISION 10 TURN POTENTIOMETER

Spectrol model 502. Resistance is 10K ± 3%. Linearity is ± .25%. 1/4" bushing mount. 1/8" shaft diameter. Body dia is .85". TTP-S10K.................. $4.50

NSL4944 LED. Current regulated, universal diffused-lens red LED lamp. A GaAsP solid-state high intensity LED encapsulated in a plastic package containing a current regulating IC that provides constant intensity over a wide voltage range. 2 to 18V, AC or DC. Use for indicator lamps, optical coupling, battery charging circuits, logic probes, almost any place you need a lamp. Long life, wide angle. No series resistor needed. Typical 13mA forward current. NSL4944.......with panel mounting clip.......... 89¢

D-A CONVERTER BY ZELTEX

8 bit precision hybrid circuit for use in controllers, timers, volt meters, etc. Molded plastic package with P.C. pins. Super buy on this better than usual subsystem. ZELTEX model ZD430. DAC-430.....................$4.95

NEW BOOK!!! "An Introduction to Microcomputers" This is the book which Fairchild Semiconductor Company called ".....the best darned introduction to the industry to date." Covers everything from basic concepts to a review of real microcomputers. IMC-001............$8.00

tri-tek, inc.
6522 north 43rd avenue,
Glendale, Arizona 85301
phone 602 - 931-6949

Minimum order $10 US/$15 Foreign in US-funds. All orders postage paid. Please add insurance. Master Charge and Bank America cards welcome, ($20 minimum) Telephone orders may be placed 11AM to 5PM daily, Mon thru Fri. Call 602-931-6949. Check reader service card or send stamp for our latest flyers packed with new and surplus electronic components.

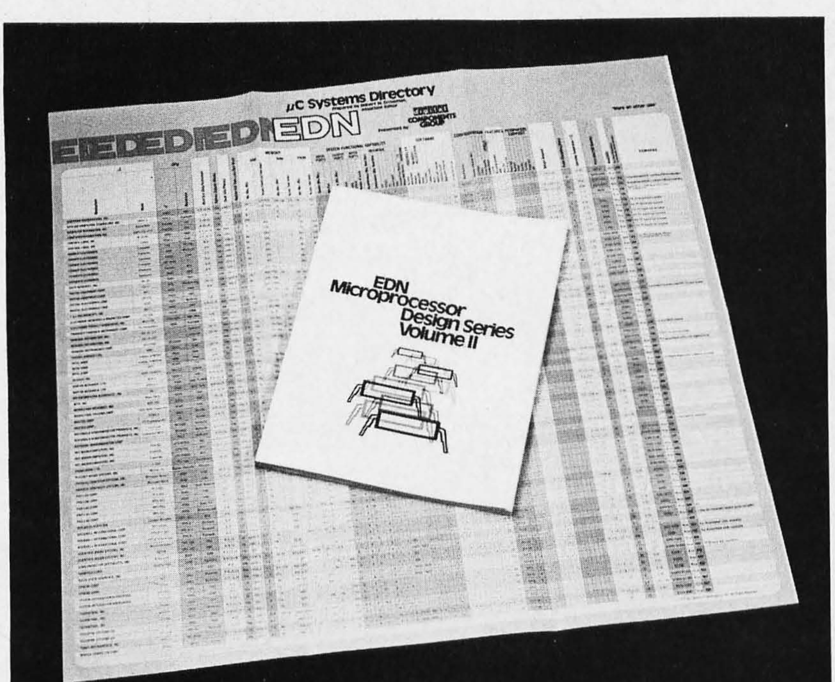
Feel free to photocopy this or any other page if you wish to keep your BYTE intact.



BOOK REVIEW

EDN Microprocessor Design Series Volume II *written and edited by EDN staff, Cahners Publishing Co, 221 Columbus Av, Boston MA 02116, 1975, $7.95 (USA) or $8.95 (foreign). A combination package of Volume I and II is available for $11 (USA) while the supply of Volume I lasts.*

To anyone who is involved in engineering design and applications involving electronics and computers, *EDN* is one of the most desirable trade magazines. [It is sometimes frustratingly difficult to get on the subscription list of this excellent publication, so a common practice at many companies is to temporarily borrow the copies whenever possible.] *EDN* has been writing about microprocessors for systems engineers virtually from the start of the technology, featuring these problem solvers in numerous technical articles. To help make the information more widely available, *EDN* came out with Volume 1 of the *EDN Microprocessor Design Series* reprints.



Volume II of the *EDN Microprocessor Design Series*, published toward the end of 1975, includes articles featured as late as November 1975. The book contains a wealth of technical information which is up to date and quite informative for the advanced computer amateur, and a must for the professionals in the audience. The orientation, as always, is toward the contemporary engineer's approach to computers and microprocessor applications. The articles are organized in four sections: "Directories and Market" summarizes the market and prices at the time of publication. This includes a very complete wall chart of 72 different vendors (underwritten by Digital Equipment Corp's Components Group) and complete lists of names and addresses of the various companies. The photo of the book accompanying this review was made with the wall chart as a backdrop. The second section is "Evaluations and Comparisons," articles on the formation of benchmark tests, trends in microprocessor technology and related subjects. The third section, "Software and Programming," is devoted to educating engineers in those arts and what to look for in instruction sets. The fourth and final section is "Design and Applications," practical details of using and applying microprocessors, including Special Features Editor Robert Cushman's series on the design and construction of the *EDN* "toy/tool" low cost microcomputer demonstration system. You'll find a complete discussion of the "toy/tool" concept in the first article and a photo story on the prototype result in a second article. Other articles are on pitfalls, the concept of a portable debugging tool, and other design topics.

For an up to date picture of the current state of the microprocessor art as viewed by one of the leading trade magazines, Volume II of the *EDN Microprocessor Design Series* is highly recommended reading.■ ...CH

Continued from page 6

can be useful for your ordinary personal correspondence. This is especially true if you use a text editor and audio tape or hard copy output as a major component of your correspondence. The addressing can then be taken right from the list, and the correspondence can be filed away in an electromagnetic duplicate for future reference. Whenever correspondence or addressing lists are involved, the computer system can be used to help reduce the burden of maintaining these activities, thus freeing your time for more enjoyable pursuits.

The applications of small computers will be enhanced by creation of standards on items of interfacing. One item which should be standardized is an 8 bit peripheral interface plug which will become the logical equivalent for small computers of what the RCA style phono plug is for audio equipment.

Enjoyment and Fascination

At the present time, the prospect of enjoyment and fascination is probably one of the most important components of the motivations to purchase and use computers. This is certainly one of my major motivations: I simply like the intricacies and complexities of programming and logical systems which allow me to program. For me, getting involved in computers is one of the great fascinations of life. But fascination devoid of goals and objectives is an empty exercise. Making unique applications is one of the major ways in which this enjoyment is realized for me, and I suspect it is a motivation for many of you as well. One of my own personal goals is the prospect of musical applications, particularly the tantalizing idea of conducting an electronic symphony orchestra controlled by the processor which acts as my baton.

This musical application is an example of the use of the computer system to accomplish new tasks previously impossible: It was never before possible for an individual of very modest means to have the artistic freedom of experimenting with a symphony orchestra. Most other applications involving enjoyment and fascination are of a similar nature: Complex activities never before possible without the computer system's automation.

Have you ever played Space War? It used to be the case that high resolution graphics and number crunching computers were out of the range of the person of ordinary means. With the new technology, we're almost to the point where a good Space War game can be had by many of the people in this audience, using ordinary TV monitors as peripherals and contemporary microcomputers as the number crunchers.

Are you a radio amateur? Then your presence at this convention is probably a reflection of your knowledge of the ways in which microcomputers can be used in that activity. At the simple end of the complexity spectrum, a microcomputer is well adapted to automation of amateur radio station activities: Simple software can be used to generate and interpret high speed precision Morse code using your ASCII keyboard and video display. Similarly, software can be easily employed to automate the conversion of Baudot to ASCII and vice versa, thus adapting your station to an FCC restriction, and incidentally allowing you to use cheap obsolete print mechanisms removed from commercial service. On a more elaborate scale, there is the prospect of an amateur radio "HARPA" network of computer automated stations for purposes of packet switching communications and nationwide time sharing of amateur computing facilities, "to prove it can be done."

Do you run a model railroad? You can have a railroad control facility far in advance of what is found on conventional commercial transportation facilities when you use your microcomputer to automate operations on your layout. The programmed control of processes is one of the biggest fields of computer application, and a model railroad is a good example of such a real time process control problem. Model railroading is an example of what might be called an "industrial modelling hobby." The same sort of fascination which infects the model railroader can also be applied to new fields with the computer automation theme. Have you ever thought of the challenge of building a model automated production line? Have you ever thought about the challenge of modelling, for instance, an automated oil refinery or pipeline system? You can do such modelling both intellectually and with actual physical hardware using the personal computer as a central element.

Another variant on the modelling hobby is that of model airplanes and other radio controlled action models. One of the latest trends in avionics is the "fly by wire" approach to aircraft control in which computers and data buses are used instead of traditional mechanical linkages and hydraulic control systems. In aircraft modelling, this approach could also be employed to some extent, as well as the use of microprocessors to implement complicated inertial navigation algorithms and pre-programmed maneuvers for the airplane which can be actuated on command of the ground station.

Are you interested in visual and graphic arts? The computer coupled with home brew plotters and other graphic output displays can be used in a number of fascinating ways to produce drawings and pictures.

Have you ever wanted to make yourself a

CRYSTALS

THESE FREQUENCIES ONLY

Part #	Frequency	Case Style	Price
CY1A	1.000 MHz	HC33/U	$4.95
CY2A	2.000 MHz	HC33/U	$4.95
CY3A	4.000 MHz	HC18/U	$4.95
CY7A	5.000 MHz	HC18/U	$4.95
CY12A	10.000 MHz	HC18/U	$4.95
CY14A	14.31818 MHz	HC18/U	$4.95
CY19A	18.000 MHz	HC18/U	$4.95
CY22A	20.000 MHz	HC18/U	$4.95
CY30B	32.000 MHz	HC18/U	$4.95

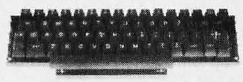


CLOCK CASES

Nicely styled cases complete with red bezel for use in such applications as desk clocks, car clocks, alarm clocks, instrument cases.
DIMENSIONS: W-4", L-4½", H-2".

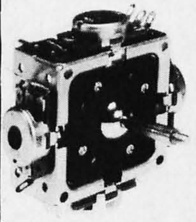
$5.95

64 Key Keyboard $24.95



This keyboard is composed of 64 Magnetic Reed Switches, in one molded unit. It is unencoded with each SPST switch brought out to two pins.

HD0165 Keyboard Encoder ROM **$7.95**



JOYSTICK

These joysticks feature four 100K potentiometers, that vary resistance proportional to the angle of the stick. Sturdy metal construction with plastics components only at the movable joint. Perfect for electronic games and instrumentation.

$9.95 ea

MOS LSI DEVICES

CLOCK CHIPS

MM5309	6 Digit, BCD Outputs, Reset PIN.	$5.95
MM5311	6 Digit, BCD Outputs, 12 or 24 Hour	4.95
MM5312	4 Digit, BCD Outputs, 1 PPS Output	4.95
MM5313	6 Digit, BCD Outputs, 1PPS Output	4.95
MM5314	6 Digit, 12 or 24 Hour, 50 or 60 Hz	4.95
MM5316	4 Digit, Alarm, 1PPS Output	6.95
MM5318	Video Clock Chip, For Use With MM5841	9.95
CT7001	6 Digit, Calander, Alarm, 12 or 24 Hour	6.95

CALCULATOR CHIPS

MM5725	6 Digit, Four Function, Less Decimal	$2.95
MM5738	8 Digit, 5 Function; +, −, ×, ÷, %	2.95
MM5739	8 Digit, 4 Function, Floating Decimal	2.95
CT5001	12 digit 4 Function	3.95
CT5005	12 digit 4 Function with Memory	5.95
CT5030	12 digit 4 Function and%	7.95

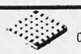
MISC. MOS

MM5320	TV Camera Sync Generator	19.95
MM5841	Video Generator For MM5318	18.00
MM5330	4½ Digit DVM Chip	9.95
MK5007	Complete 4 Digit Counter	$10.95
LD110/LD111	3½ Digital Voltmeter Chip Set	$25.00
MC1408L7	7 Bit Digital To Analog Conv.	9.95

DL728

The DL728 is a dual 0.5" common cathode red display. It is ideal for use with clock chips, as segments are already multiplexed. **$2.95**





1/16 VECTOR BOARD

0.1" Hole Spacing P-Pattern

	Part No.	Length	Wideth	Price 1-19	Price 20-49
PHENOLIC	64P44 062XXXP	4.50	6.50	1.72	1.54
	169P44 02XXXP	4.50	17.00	3.69	3.32
EPOXY GLASS	64P44 062	4.50	6.50	2.07	1.86
	84P44 062	4.50	8.50	2.56	2.31
	169P44 062	4.50	17.00	5.04	4.53
	169P84 062	8.50	17.00	9.23	8.26
EPOXY GLASS COPPER CLAD	169P44 062C1	4.50	17.00	6.80	6.12

VECTOR WIRING PENCIL



Vector Wiring Pencil P173 consists of a hand held featherweight (under one ounce) tool which is used to guide and wrap insulated wire, fed off a self-contained replaceable bobbin, onto component leads or terminals installed on pre-punched "P" Pattern Vectorbord*. Connections between the wrapped wire and component leads, pads or terminals are made by soldering Complete with 250 FT of red wire. **$9.50**

REPLACEMENT WIRE — BOBBINS FOR WIRING PENCII

W36-3-A-Pkg. 3	250 ft. 36 AWG GREEN	$2.40
W36-3-B-Pkg. 3	250 ft. 36 AWG RED	$2.40
W36-3-C-Pkg. 3	250 ft. 36 AWG CLEAR	$2.40
W36-3-D-Pkg. 3	250 ft. 36 AWG BLUE	$2.40

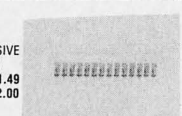
1¼" x 1½" XFMERS — P.C. Mount

That were designed for clock type applications. 110 Vac primary @ 60 Hz.
Secondaries 8-10 Vac @ 30 mA-50 mA
50 Vac @ 30 mA-50 mA
Excellent for miniature power supplies & gas discharge displays

SPECIAL $.79

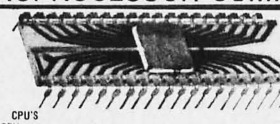


MOLEX PINS
PINS USED AS INEXPENSIVE SUBSTITUTE FOR SOCKETS
SPECIAL 100/1.49
SPECIAL 1000/12.00



MICROPROCESSOR COMPONENTS

8080 $24.95



8080A $39.95

CPU'S			RAM'S			
8008	8 Bit CPU	$19.95	1101	256X1	STATIC	$2.25
8080	Super 8008	$24.95	1103	1024X1	DYNAMIC	2.95
8080A	Super 8008	$39.95	2101	256X1	STATIC	6.95
			2102	1024X1	STATIC	2.49
SR'S			2107	4096X1	DYNAMIC	19.95
2504	1024 DYNAMIC	$9.00	2111	256X4	STATIC	7.95
2518	HEX 32 BIT	7.00	7010	1024X1	MNOS	29.95
2519	HEX 40 BIT	4.00	7489	16X4	STATIC	2.49
2524	512 DYNAMIC	2.95	8101	256X4	STATIC	7.95
2525	1024 DYNAMIC	6.00	8111	256X4	STATIC	7.95
2527	DUAL 256 BIT	3.95	8599	16X4	STATIC	3.49
2529	DUAL 512 BIT	4.00	91L02	1024X1	STATIC	2.75
2532	QUAD 80 BIT	3.95	74200	256X1	STATIC	6.95
2533	1024 STATIC	7.95	93410	256X1	STATIC	1.75
3341	FIFO	6.95	5262	2048X1	DYNAMIC	2.95
74LS670	16X4 REG	3.95				
UART'S			**PROMS**			
AY-5-1013	20K BAUD	$6.95	1702A	2048	FAMOS	15.95
ROM'S			5203	2048	FAMOS	14.95
2513	CHAR. GEN.	11.00	82S23	32X8	OPEN C.	5.00
7488	RANDOM BITS	3.50	82S123	32X8	TRISTATE	5.00

BUILD YOUR OWN JOLT MICROCOMPUTER IN JUST 3 HOURS OR LESS FOR $159.95.

A COMPLETE MICROCOMPUTER IN A SINGLE CPU KIT INCLUDES: • An MOS Technology MCS6502 NMOS microprocessor • 512 bytes of program RAM, and 64 bytes of interrupt vector RAM • 1K bytes of mask programmed ROM containing DEMON, a powerful debug monitor • 26 programmable I/O lines • Internal RC clock, or crystal controlled clock with user supplied crystal • Serial I/O ports for use with a teleprinter current loop drive/receiver, or an EIA standard driver/receiver • Expandable address and data buses • Hardware interrupt • Control panel interface lines available on card connector • Complete assembly manuals and sample programs

JOLT ACCESSORY KITS

JOLT RAM Card — Fully static 4,096 bytes of RAM with 1 microsecond access time and on-board decoding **$199.95**

JOLT I/O Card (Peripheral Interface Adapter) — 2 PIA LSI chips, 32 I/O lines, four interrupt lines, on-board decoding and standard TTL drive. Fully programmable. **$95.50**

JOLT Power Supply — Operates at +5, +12 and −10 voltages. Supports JOLT CPU, 4K bytes of RAM and JOLT I/O card — or, CPU and 8 I/O cards. **$99.95**

JOLT +5V Booster Option — Fits onto JOLT Power Supply card. Supports CPU, 8K bytes RAM and 8 I/O CPU and cards. **$24.95**

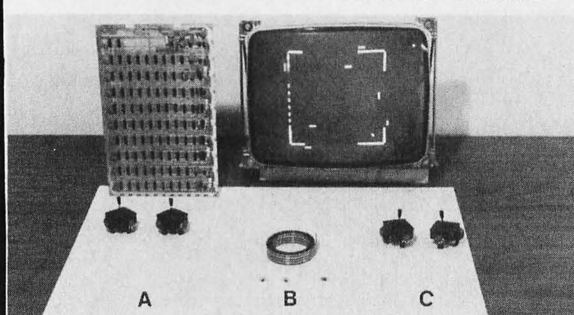
JOLT Universal Card — Same size (4¼" x 7"), same form factor as other JOLT cards. Completely blank, drilled to accept 14, 16, 24 or 40 pin sockets. **$24.95**

JOLT Accessory Bag — Contains enough hardware to connect one JOLT card to another, flat cable, connectors, card spacers, hardware, etc. **$39.95**

JOLT Resident Assembler — Fully symbolic, single pass resident assembler, all mnemonics compatible with timesharing assemblers. Delivered on four 1702A PROMs, ready for plugging into JOLT PROM card. **$149.95**

JOLT 1702A PROM Card — Sockets for 2,048 bytes of PROM memory. Place anywhere in memory with jumper selectable addresses. **$99.95**

NOW: YOUR OWN VIDEO GAME FOR THE ENTIRE FAMILY



This game comes pre-tested with two **PROFESSIONAL** Kraft joysticks. Joysticks allow 2 dimensional player control (rather than only one dimension, such as up and down.) If you require more than two players, order extra joysicks. All that's required is a 5V/2A power supply, a harness, and speaker. Comes with schematics, wiring information, and all necessary documentation, Game gives 1V composite video output, perfect for any TV monitor, Game designed so one, two, three, or four players can play at the same time. You can even play against the HOUSE. Score for each person is shown on TV set. These boards are production over runs of a well known video game manufacturer, and are **not** rejects, or in any way inferior to one presently being sold in games for over $1,000.00.

KIT A — $179.95 **PROFESSIONAL** game P.C. board, and 2 **PROFESSIONAL** joysticks. P.C. board size is 10½" x 17". This is the same **PROFESSIONAL** game as seen in commercial establishments. Don't confuse it with the simple games sold in stores, or with analog kits.

ACCESSORY B — $3.95 Six feet of ribbon cable, three SPST switches (coin simulator, house/player, and start switch), for those of you who don't have extra wire or switches to build the harness.

ACCESSORY C — $39.95 Two additional **PROFESSIONAL** Kraft joysticks, for third and fourth player.

DIGITAL CLOCK KIT — 3½ INCH DIGITS



This clock features big 3½" high digits for viewing in offices, auditoriums, etc. Each digit is formed by 31 bright 0.2" LED's. The clock operates from 117 VAC, has either 12 or 24 hr. operation. The 6 digit version is 27" x 3½" x 1½", and the 4 digit is 18" x 3½" x 1½". Kits come complete with all components, case and transformer.
Specify 12 Or 24 Hr. When Ordering

4 DIGIT KIT $49.95 **4 DIGIT ASSEMBLED $59.95**
6 DIGIT KIT $69.95 **6 DIGIT ASSEMBLED $79.95**

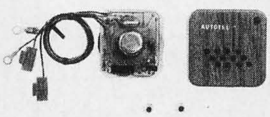
Satisfaction Guaranteed. $5.00 Min. Order. U.S. Funds.
California Residents — Add 6% Sales Tax — Data Sheets 25c each
Send a 13c Stamp (postage) for a FREE 1976 Catalog

JAMES

P.O. BOX 822, BELMONT, CA. 94002
PHONE ORDERS — (415) 592-8097

AUTOTEL ™

WARNING LIGHTS ARE NOT ENOUGH



AN AUDIBLE ALARM INDICATING POTENTIAL ENGINE DAMAGE

AUTOTEL is an effecient (15 ma current standby) device by which every owner of an automobile, truck or vehicle equipped with indicator lights for temperature and oil pressure can be assured of a reliable warning **before** an impending failure.

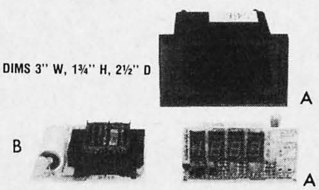
AUTOTEL, by means of an audible signal 70 db pulsing) immediately forewarns the vehicle operator of a malfunction or failure, allowing time to correct and prevent major engine failures. It is programmed so there is no sound during normal operating conditions.

AUTOTEL features CMOS circuitry, packaged in a 2½" sq. x ⅞" case. The kit comes complete with all components, hardware and case to hook directly into your car's warning light system.

$9.95 Per Kit **$14.95 Assembled**

IMC 3½ DIGIT DVM KIT

DIMS 3" W, 1¾" H, 2½" D



This 0-2 VDC .05 per cent digital voltmeter features the Motorola 3½ digit DVM chip set. It has a .4" LED display and operates from a single +5V power supply. The unit is provided complete with an injection molded black plastic case complete with Bezel. An optional power supply is available which fits into the same case as the 0-2V DVM allowing 117 VAC operation.

A. 0-2V DVM with Case $49.95
B. 5V Power Supply $14.95

JE700 CLOCK

The JE700 is a low cost digital clock, but is a very high quality unit. The unit features a simulated walnut case with dimensions of 6" x 2½" x 1". It utilizes a MAN72 high brightness readout, and the MM5314 clock chip.



12 or 24 Hour

115 VAC— $17.95 Per Kit

JE803 PROBE

The Logic Probe is a unit which is for the most part indespensible in trouble shooting logic families. TTL, DTL, RTL, CMOS. It derives the power it needs to operate directly off of the circuit under test, drawing a scant 10 mA max. It uses a MAN3 readout to indicate any of the following states by these symbols: (H) - 1 (LOW) - o (PULSE) - P. The Probe can detect high frequency pulses to 45 MHz. It can't be used at MOS levels or circuit damage will result.



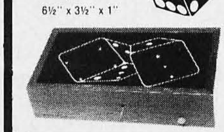
$9.95 Per Kit
printed circuit board

ELECTRONIC CRAPS



6½" x 3½" x 1"

Electronic craps is an entirely electronic game featuring a dice "roll down". 14 LED's form two dice that roll when activated by a push botton. Dimensions are 6½" x 3½" x 1½".



$19.95 Per Kit
$24.95 Assembled

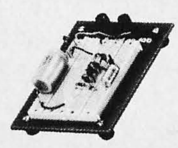
ELECTRONIC ROULETTE



This kit comes complete with all components, including the case and line cord. Electronic Roulette, is entirely electronic, and features 32 LED's that form a wheel, that is activated, by a push botton switch. Dimensions are 6½" x 6½" x 1½".

$29.95 Per Kit
$34.95 Assembled

Continental Specialties



PROTO BOARD 100
A low cost, big 10 IC capacity breadboard kit with all the quality of QT sockets and the best of the Proto Board series . . . complete down to the last nut, bolt and screw. Includes 2 QT-35S Sockets; 1 QT-35B Bus Strip; 2 5-way binding posts; 4 rubber feet; screws, nuts, bolts; and easy assembly insturctions.

$19.95

robot? It is virtually impossible to consider the problem of making your own robot mechanisms without a computer to control the show and calculate where the arms, legs, and other mechanisms should be going based upon inputs from the environment. I know of several people who are already working on robots as a hobby, and expect that this will be one of the most exciting and challenging subsets of the activity of personal computing experimenters.

New Technology and Practical Applications

The new technology of personal computing of course provides numerous practical applications which could not previously have been accomplished. These tasks require a computer.

How about automation of your household? Start by using the computer as a keystone of an advanced burglar and fire processing system. For the time being, since computer technology is so far in advance of the norm, the element of surprise and befuddlement which is necessary to scare off a burglar can be greatly augmented through computer approaches. These can range from improved sensor monitoring and discrimination, to improved methods of making the burglar nervous and uncomfortable in his activities. As a sensor of fires, the sophistication of the alarm system can be improved and it might even be possible to provide selective automatic extinguishing operations depending upon the nature of the fire sensed.

Don't stop at the defensive mechanisms of your household. Continue your automation into areas of cost cutting and efficiency. Have your household utility computer monitor fuel flow, control the zone controls of your heating system based on occupancy of the rooms in question, and incorporate cost effectiveness calculations into your heating and air conditioning systems where there are options. For instance, with a very simple BASIC program, you could instantly evaluate the proper mix of wood furnace versus oil furnace versus electric base board heating in a typical New England household which has all three options installed.

The day of the ROM chip rack in a retail store, or a weekly software special on cassette tape may not be far away.

Then there is the new technology of inexpensive text manipulation and storage. At the surface, this is a straightforward application. But there is the prospect of ultimate privacy for your personal records as well, since the computer can be used to automate ciphers which require extreme amounts of time on large scale computers to crack, but which are simple to translate given the key of which you are the sole possessor. Thus you are the sole deciding element as to who sees the records under normal circumstances, and extraordinary circumstances become extremely costly for the person or agency attempting access to your files.

Besides the security element of ciphers used for personal files, the very prospect of advanced information processing at home is exciting. For instance, we ran an article in BYTE on the "Total Kitchen Information System," which is a subset of the various filing and information retrieval tasks which could be aided by the personal computer at home. The text processing computer can be used to automate kitchen recipe files, little black books for bachelors, business contact lists, correspondence, book manuscripts, school research papers, and a host of other problems where character text is a central component.

What are the Prognostications?

So much for the observations and imagination inputs; what will the technological trends be, as these personal uses of computers expand? How will the present small industry of personal computing manufacturers change and expand as time goes on? In short, what are the prognostications for the future?

Product Evolution

The trend in product marketing and packaging for the near future is "smoothing out the wrinkles." Present day kits work well, and can be mastered by many individuals, as demonstrated by the many people who have bought and are using them. However, with no attempt at being unjustifiably critical, present day computer kits are in many respects like experimental aircraft kits. A bit of sophistication is often necessary on the part of purchasers. But people buy, build and fly experimental aircraft, and people buy, build and program present day computers in kit form. The next step in product evolution for this market will be the "computer system component" product analogous to the high fidelity market. Here the idea is that a user purchases an assembled product, not a kit, and with a minimum of trouble interfaces this product to other products, not necessarily of the same manufacturer.

Standards

In order to achieve newer and wider markets composed of individuals who are not handy with soldering iron and oscilloscopes, there are several industry wide considerations in the area of "standardization." Just as railroads soon found the advantages of standard track gauges, automotive manu-

SOLID STATE SALES

Features Value

SPECIAL PURPOSE TRANSISTORS

INTEGRATED CIRCUITS

HARDWARE + P.C. SOCKETS

SILICON POWER RECTIFIERS

ZENERS

TRIACS

SCRS

LIGHT EMITTING DIODES

SPECIAL PURCHASE ITEMS

WE HAVE AVAILABLE

USED IBM SELECTRIC 731 INPUT/OUTPUT TYPEWRITERS,

WITH MANUALS FOR $350.00

ALSO AVAILABLE USED KLEINSCHMIDT M-311 PRINTERS

THEY POINT AT THE RATE OF 120 CHARACTERS/SECOND. $450.00

VIDEO CAMERA KIT

A UNIQUE ALL SOLID STATE CAMERA KIT FEATURING A . . . 100 X 100 BIT SELF SCANNING CHARGED COUPLED DEVICE

INCLUDES THE FOLLOWING UNIQUE FEATURES . . . FOUND IN FAR MORE EXPENSIVE CAMERAS IF AVAILABLE

- LOW VOLTAGE SUPPLY (OR BATTERIES) + 5 AND ± 15 VOLTS
- SENSITIVE TO INFRA RED AS WELL AS VISIBLE LIGHT
- MAY BE USED FOR IR SURVEILLANCE WITH AN IR LIGHT SOURCE
- EXCELLENT FOR STANDARD SURVEILLANCE WORK BECAUSE OF ITS LIGHT WEIGHT AND SMALL SIZE
- ALL COMPONENTS MOUNTED ON TWO PARALLEL 3" X 5" SINGLE SIDED BOARDS TOTAL WEIGHT UNDER 2 LBS
- MAY BE WIRED BY PERSON WITH SOME TECHNICAL EXPERIENCE IN 4-6 HRS



SUPER UNBELIEVABLE

ONLY $225.00

MAY BE USED WITH AMATEUR RADIO FOR VIDEO

USED FOR CHARACTER RECOGNITION FOR COMPUTERS WITH EXTERNAL CIRCUITS

MAY BE USED IN A VACUUM, UNDER WATER, HIGH ALTITUDE, AND IN MAGNETIC ENVIRONMENT BECAUSE THERE IS NO HIGH VOLTAGE OR MAGNETIC DEFLECTION

WE SUPPLY ALL SEMICONDUCTORS, BOARDS, DATA SHEETS, DIAGRAMS, RESISTORS AND CAPACITORS.

SORRY, WE DO NOT SUPPLY THE CASE, BATTERIES (OR SUPPLY) THE LENSE (NOT SUPPLIED) DEPENDS UPON THE USE

ADD $2 00 POSTAGE AND HANDLING. PLUS $6 00 FOR ALL IC SOCKETS (Optional)

F8 MICROPROCESSOR KIT

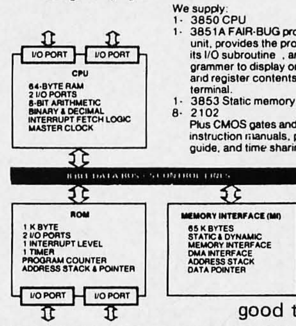
WE'VE GOT THE F8 MICROPROCESSOR KIT, ONE OF THE MOST ADVANCED MCU SYSTEMS ON THE MARKET TODAY FOR ONLY **$179.00**

This three chip microprocessor system has the following advantages:

1) Driven by a +5 and +12 volt power supply
2) Two I/O ports on the CPU chip, and ROM, making 32 bidirectional lines.
3) 64 bytes of fast RAM scratchpad built into the CPU chip.
4) A built in clock generator and power-on reset built into the CPU chip.
5) A programmable internal timer built into the ROM chip
6) 60% of the instructions are 1 byte
7) TTL I/O compatibility.
8) consumes less than 300mw of power per chip
9) A local interrupt with automatic adress vector.
10) expandable to 64K bytes (2102-1's) of memory.
11) 20 mil loop & RS – 232 included.

The F8 Kit has enough parts and instructions to demonstrate microprocessor programs up to 1K byte, and to debug those programs.

We supply:
1- 3850 CPU
1- 3851A FAIR-BUG programmed storage unit, provides the programmer with all its I/O subroutine , and allows the programmer to display or alter memory, and register contents via a teletype terminal.
1- 3853 Static memory interface
8- 2102
Plus CMOS gates and buffers, PC card, instruction manuals, programming guide, and time sharing guide.



good to 1 MHZ

INTEL 8080 CPU	$37.50
8008 8 BIT MICRO PROCESSING CHIP (with Data Book)	$19.00
2102-1 1024 BT RAM	$ 2.60
5202A UV PROM	$12.50
MM5203 UV PROM	$12.50
1702A UV PROM	$12.50
5204-4K PROM	$24.95

MINATURE MULTI-TURN TRIM POTS 100, 500, 5K, 10K, 25K, 50K, 100K, 200K
$.75 ea. 3/$2.00

MULTI-TURN TRIM POTS Similar to Bourns 3010 style 3/16" x 5/8" x 1 1/4"; 50, 100, 1K, 10K 50K ohms
$1.50 ea. 3/$4.00

LIGHT ACTIVATED SCR's TO-18, 200V 1A **$1.75**

PRINTED CIRCUIT BOARD

4 1/2" x 6" SINGLE SIDED EPOXY BOARD 1/16" thick, unetched
$.50 ea. 5/$2.20

VECTOR BOARD 1" SPACING 4.5" x 6.5" SHEET **$1.50**

4 WATT IR LASER DIODE $7.95

2N 5460 P FET	$ .45
2N 5457 N FET	$ .45
2N 4891 UJT	$ .45
TIS 43 UJT	$ .35
ER 900 TRIGGER DIODES	4/$1.00
2N 6028 PROG. UJT	$ .65

VERIPAX PC BOARD

This board is a 1/16" single sided paper epoxy board, 4 1/2" x 6 1/2", DRILLED and ETCHED which will hold up to 21 single 14 pin IC's or 8, 16, or LSI DIP IC's with busses for power supply connector. $5.25

MV 5691 YELLOW-GREEN BIPOLAR LED	$1.25
MT-2 PHOTO TRANS.	$ .60
RED, YELLOW, GREEN OR AMBER LARGE LED's	ea. $ .20
14 PIN DIP SOCKETS	$ .35
16 PIN DIP SOCKETS	$ .38
MOLEX PINS	100/$1.00
	1000/$8.00
8 PIN MINI DIP SOCKETS	$ .30
10 PIN TO-5 TEFLON PC SOCKETS	$ .60
10 WATT ZENERS 3.9, 4.7, 12, 18 OR 22V	ea. $ .60
1 WATT ZENERS 4.7, 5.6, 10, 12, 15, 18 OR 22V	ea. $ .25

Silicon Power Rectifiers

PRV	1A	3A	12A	50A	125A
100	.06	.14	.30	.80	
200	.07	.20	.35	1.15	4.25
400	.09	.25	.50	1.40	6.50
600	.11	.30	.70	1.80	8.50
800	.15	.35	.90	2.30	10.50
1000	.20	.45	1.10	2.75	12.50

REGULATED MODULAR POWER SUPPLIES

+ – 15VDC AT 100ma 115VAC INPUT	$24.95
5VDC AT 1A. 115VAC INPUT	$24.95
12VDC AT .54	$24.95
IN 4148 (IN914)	15/$1.00
1103, 1024 bit RAM	$2.75
NEC 6003 2048 bit RAM	$6.00
1101 256 bit RAM	$1.40
7489 RAM	$2.30
7 POLE 1 THROW TO-5 MINATURE ROTARY SWITCH	$ .95

TRANSISTOR SPECIALS

2N3585 NPN Si TO-66	$ .95
2N3772 NPN Si TO-3	$1.80
2N4901 PNP Si TO-3	$ .85
2N5086 PNP Si TO-92	4/$1.00
2N4898 PNP TO-66	$ .60
2N404 PNP GE TO-5	5/$1.00
2N3919 NPN Si TO-3 RF	$1.50
MPSA13 NPN Si TO-92	3/$1.00
2N3767 NPN Si TO-66	$ .70
2N2222 NPN Si TO-18	5/$1.00
2N3055 NPN Si TO-3	$ .80
2N3904 NPN Si TO-92	5/$1.00
2N3906 PNP Si TO-92	5/$1.00
2N5296 NPN Si TO-220	$ .50
2N6109 PNP Si TO-220	$ .55
2N3866 NPN Si TO-5	$ .75
2N3638 NPN Si TO-5	5/$1.00
2N6517 NPN TO-92 Si	3/$1.00

TANTULUM CAPACITORS

.22UF 35V	5/$1.00	6.8UF 35V	3/$1.00
.47UF 35V	5/$1.00	33UF 25V	$ .40
.68UF 35V	5/$1.00	30UF 6V	.5/$1.00
1UF 35V	5/$1.00	150UF 20V	$ .50
4.7UF 35V	4/$1.00		

CT7001 ALARM CLOCK CHIP	$5.75
FPA 711-7 LEVEL Diode Array Optical Tape Readers	$5.95

NATIONAL MOS DEVICES

MM1402—3.20	MM5057—4.00
MM1403—3.20	MM5058—4.95
MM1404—2.50	MM5060—4.95
MM5013—7.75	MM5061—4.30
MM5016—3.50	MM5555—6.25
MM5017—4.75	MM5556—6.25
MM5055—4.00	MM5210—1.95
MM5056—4.00	MM5260—2.95

TTL IC SERIES

74L00— .26	7476— .35
7400— .14	7480— .48
7401— .15	7483— .75
7402— .15	7485—1.05
7403— .15	7486— .34
7404— .18	7490— .49
7405— .18	7491— .79
7406— .35	7492— .49
7407— .33	7493— .49
7408— .18	7495— .74
7409— .18	7496— .79
7410— .15	74107— .34
7411— .25	74121— .38
7412— .30	74123— .74
7413— .45	74125— .54
7414—1.45	74126— .54
7416— .33	74150— .92
7417— .33	74151— .69
7420— .15	74153— .79
7425— .27	74154—1.05
7426— .24	74155—1.05
7427— .27	74157— .75
7430— .15	74161— .95
7432— .24	74164—1.10
7437— .29	74165—1.10
7438— .29	74173—1.35
7440— .15	74175— .95
7441— .95	74177— .75
7442— .59	74181—2.30
7445— .79	74192—1.10
7446— .87	74193—1.10
7447— .87	74194—1.15
7448— .85	74195— .74
7472— .28	74196— .95
7473— .34	75324—1.75
7474— .35	75491—1.10
7475— .49	

ALCO MINATURE TOGGLE SWITCHES

MTA 106 SPDT	$1.20
MTA 206 DPDT	$1.70

C/MOS (DIODE CLAMPED)

74C02— .26	4013— .45	4028— .95
74C10— .30	4015—1.24	4029—1.20
4001— .24	4017—1.15	4030— .49
4002— .24	4018—1.24	4035—1.50
4006—1.35	4019— .50	4042— .75
4007— .24	4022—1.24	4046—2.75
4009— .50	4024— .95	4047—3.50
4010— .50	4025— .24	4050— .49
4011— .24	4026—1.90	4055—1.95
4012— .24	4027— .55	4066—1.00

LED READOUTS

FND 500-.5" C.C.	$1.75
HP 7740-.3" C.C.	$1.40
MAN-4-.25" C.C.	$1.20
MAN-7-.3" C.A.	$1.25
DL 747-.6" C.A	$1.95
NS 33-3 dig. array	$1.35

Full Wave Bridges

PRV	2A	6A	25A
200	.95	1.25	2.00
400	1.15	1.50	3.00
600	1.35	1.75	4.00

CD 110 LINEAR 256 XI BIT SELF SCANNING CHARGED COUPLED DEVISE . . . **$125.00**

SANKEN AUDIO POWER AMPS

Si 1010 G 10 WATTS	$ 6.90
Si 1020 G 20 WATTS	$13.95
Si 1050 G 50 WATTS	$24.95

LINEAR CIRCUITS

LM 309K 5V 1A REGULATOR	$1.00
723 — 40 + 40V REGULATOR	$ .54
301/748-Hi Per. Op. Amp.	$ .28
320T 5, 12, 15, OR 24V NEGATIVE REG.	$1.50
741A or 741C OP. AMP.	$ .31
709c OPER. AMP	$ .25
307 OP AMP	$ .25
CA 3047 HI PERF. OP AMP	$ .95
CA 3089 FM IF SYSTEM	$3.25
340T-5, 8, 12, 15, 18, 24V POS. REG. TO-220	$1.20
101 OPER. AMP. HI PERFORM	$ .75
LM 308 Oper. Amp., Low Power	$1.05
747 — DUAL 741	$ .65
556 — DUAL TIMER	$ .95
537 — PRECISION OP. AMP	$2.60
540-70W POWER DRIVER	$2.50
LM 3900 — QUAD OP. AMP	$ .49
LM 324 — QUAD 741	$1.50
560 — PHASE LOCK LOOP	$2.50
561 — PHASE LOCK LOOP	$2.50
565 — PHASE LOCK LOOP	$1.50
566 FUNCTION GEN.	$2.50
567 — TONE DECODER	$1.50
LM 1310N FM STEREO DEMOD	$2.75
8038 IC VOLTAGE CONT. OSC.	$3.90
LM370 — AGC SQUELCH AMP.	$1.15
555 — 2 μs — 2 HR. TIMER	$ .53
553 QUAD TIMER	$2.50
FCD 810 OPTO-ISOLATOR	$ .80
1458 DUAL OP. AMP.	$ .55
LM 380 — 2W AUDIO AMP.	$ .80
LM 377 — 2W Stereo Audio Amp.	$2.50
LM 381 — STEREO PREAMP	$1.00
LM 382 — DUAL AUDIO PREAMP	$1.00
LM 311 — HI PER. COMPARATOR	$ .95
LM 319 — Dual Hi Speed Comp.	$1.10
LM 339 — QUAD COMPARATOR	$1.25

TRIACS / SCR'S

PRV	TRIACS 1A	TRIACS 10A	TRIACS 25A	SCR'S 1.5A	SCR'S 6A	SCR'S 35A
100	.40	.70	1.30	.40	.50	1.20
200	.70	1.10	1.75	.60	.70	1.60
400	1.10	1.60	2.60	1.00	1.20	2.20
600	1.70	2.30	3.60		1.50	3.00

Send 25¢ for our catalog featuring Transistors and Rectifiers
145 Hampshire St., Cambridge, Mass.

Terms FOB Cambridge, Mass
Send Check or Money Order
Include Postage Minimum
Order $5.00, COD'S $20.00

SOLID STATE SALES
P.O. BOX 74B
SOMERVILLE, MASS. 02143 TEL. (617) 547 4005

WE SHIP OVER 95% OF OUR ORDERS THE DAY WE RECEIVE THEM

An audio turntable with cuing controls is the only presently available "poor man's disk drive," should manufacturers and other organizations choose to distribute audio encoded information on long playing records.

facturers found the advantages of standard tire sizes, and electronics manufacturers use standardized integrated circuit parts, so too there is a definite need for standardization in several areas of the personal computing industry if we are to achieve the convenience of interconnection of components which characterizes the modern high fidelity audio system.

A step in the direction of standardization began when BYTE magazine sponsored an audio recording standards conference last November, a conference which resulted in selecting the provisional standard described in BYTE's February and March issues. As a continuation of this trend, BYTE is proposing a second conference next fall to discuss experiences with the recording format, including software data formats, as an addition to the standard. Also needed is the specification of a standard 8 bit peripheral interface plug standard, which will become the logical equivalent of the RCA style phono plug used in audio equipment. There is a proliferation of central processor designs on the market, so much so that the maker of an applications oriented peripheral such as a music synthesizer black box, a burglar alarm black box, a TV graphics generator, a hard copy printer, a floppy disk system, or the like has no way to ensure that his black box will plug into everybody's computer regardless of manufacturer. By providing a standard IO plug for 8 bit data quanta with interrupt and directional control as well as strobes, the industry can be expanded considerably. The more options people have with their personal computers, the more desirable is the product.

Software Markets?

Another area which should be developed is the software markets. It is not clear yet what this means. One item which we'll be adding to BYTE at some point in the future is a commercially oriented classified software advertisement which will enable small operations to economically market software packages. It is fairly obvious that in this personal use market, with many individuals on limited budgets, one aspect of software which must be considered is price and mass production. A high fidelity record, for instance, is mass marketed with a small but nagging incidence of piracy. If the price of a stereo record were to double, the instances of copying and piracy would go even higher. By analogy, software prepared for the personal use markets will not be salable at high prices in the same way that large computer software packages are sold at high prices. The problem of contract enforcement may become intractable with large numbers of users.

The fact of the matter is that people of low moral character have few scruples about copying a work which takes time, money and effort to develop. One answer to the problem is to not write software at all. Another answer is to mass produce applications software at low prices so that software becomes the personal systems industry equivalent of a high fidelity record. In the record industry, most people simply go out and purchase the record, thus crediting the artists and recording company with the royalty. The reason is that if you add up all the costs – both moral and economic – the difference in price between a legitimate recording and a pirated one is miniscule, or in favor of the legitimate recording. It is not clear yet how such mass production will work out in detail, but the day of the ROM chip rack in a retail store, or a weekly software special on a cassette tape may not be far away. It is quite likely that most of the software being sold at retail in this way will be for application packages which are written using software development tools of limited distribution. The packages of widest market will most likely be relocatable object code for particular applications on particular microcomputers; assemblers, interpreters and compilers may never become more than bundled packages distributed by manufacturers.

Mass Storage Trends

The present day situation in mass storage methods for personal computing is fairly primitive. The only widely distributed mass storage method is audio and direct digital recording on inexpensive cassette or reel to reel recording devices. These methods are reliable and accomplish the goal of electromagnetic off line recording, but they fall far short of the random access requirement needed for a good information storage and retrieval system.

In order to fully accomplish the convenient personal computing function, there is a definite market need for an inexpensive random access mass storage system. About the only way this can be done inexpensively right now is through the medium of phonograph record technology applied to read only copies of software packages represented as audio recordings of digital information. The placement of the arm of the phonograph on a particular band of the record, using a cuing control, is a poor man's equivalent of a disk drive access arm.

Here is what is needed of inventors and manufacturers: a budget version of the



VIATRON terminal. Unused, consists of keyboard, micro-processor, control panel, video display, 2 built-in tape decks, power supply. Operates on 115v AC 60 cycle. Unused but in storage for 4 years. Due to storage, may require some work. Sold "as is" FOB Lynn, Mass. Ship wgt. 160 lbs. Has memory, automatic input/output under program control, tape search, key verification, tape validation, etc.

Send for informative data brochure. Limited quantity. $425.00 FOB Lynn, Mass.

CLOCK KIT $14.00

Includes all parts with MM5316 chip, etched & drilled PC board, transformer, everything except case.

SP-284 $14.00 each 2/$25.00



ASCII KEYBOARD, brand new w/TI ASCII chip inplace & data $45.00

COMPUTER GRADE LOGIC SUPPLY CAPS, BRAND NEW

47,000 Uf	**25V**	**$2.00**	**ST**	**1,000**	**50**	**.90**	**AL**
32,000	**25**	**1.75**	**ST**	**3,300**	**35**	**1.25**	**AL**
160,000	**10**	**2.00**	**ST**	**1,600**	**20**	**.60**	**AL**
66,000	**10**	**2.00**	**ST**	**8,000**	**16**	**1.25**	**AL**
1,000	**60**	**.90**	**AL**	**500**	**6**	**.35**	**AL**
2,000	**55**	**1.00**	**AL**	**"ST" screw top . . . . "AL" axial**			

5 VOLT 1 AMP REGULATED power supply kit for logic work. All parts including LM-309K **$7.50**

DUMMY LOAD RESISTOR, non inductive, 50 ohm 5 watt **$1.00**

"AA" NICAD CELLS brand new, fine biz for handy talkies **$1.25 ea. 9/$9.00**

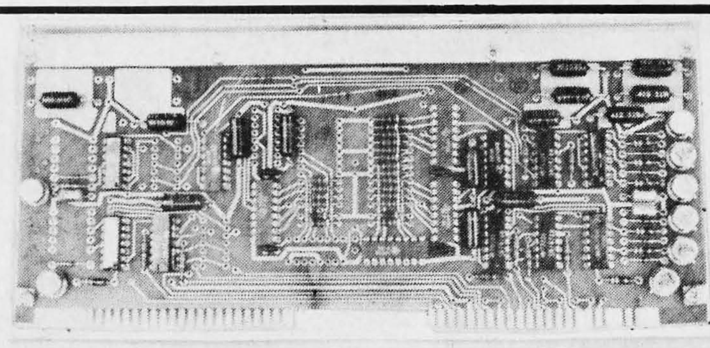
LINEAR by RCA, brand new, gold bond process

301	$ .60	747	$ .82	MM5314	$3.00
307	.52	748	.50	MM5316	3.00
324	1.80	1458	.96	7001	8.00
339A	1.60	3401	.80		
741	.50	555 timer	.60		

MEMORY SYSTEM $125.00

New memory system by Honeywell, small ... measures only 9x4x1 inches. 1024 core memory, 1024 words with 8,9,10 bits/word. Random access, with all logic, register, timing, control, core select and sense functions in one package. New, booklet of schematics and data. Looks like a good beginning for a mini-computer. Limited supply on hand.

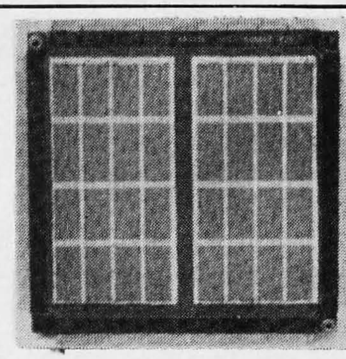
Ship wgt 3 lbs. #SP-79 . . . . $125.00



CORE MEMORY

Another brand new memory, ultra small. Measures only 4 x 4 inches with format on one plane of 32 x 32 x 16 (16,384). Only about 35 units of this on hand.

#SP-81 . . . . $20.00



Meshna

FREE CATALOG SP-7 NOW READY

Please add shipping cost on above.

MESHNA PO Bx 62 E. Lynn Mass. 01904

BOMB: BYTE's Ongoing Monitor Box

BYTE would like to know how readers evaluate the efforts of the authors whose blood, sweat, twisted typewriter keys, smoking ICs and esoteric software abstractions are reflected in these pages. BYTE will pay a $50 bonus to the author who receives the most points in this survey each month.

- *Articles you like most get 10 points, articles you like least get 0 (or negative) points – with intermediate values according to your personal scale of preferences, integers only.*
- *Only one entry per reader.*

Page No.	Article	LIKED LEAST ... BEST
8	Simpson: A Date With KIM	0 1 2 3 4 5 6 7 8 9 10
18	Boudinot: n Source	0 1 2 3 4 5 6 7 8 9 10
24	Eichbauer: ROM in uP Memory Address Space	0 1 2 3 4 5 6 7 8 9 10
28	Smith: More Information on PROMs	0 1 2 3 4 5 6 7 8 9 10
36	Haller: Serialize the Bits . . .	0 1 2 3 4 5 6 7 8 9 10
38	DeMonstoy: An Octal Front Panel	0 1 2 3 4 5 6 7 8 9 10
42	Nico: SHOOTING STARS	0 1 2 3 4 5 6 7 8 9 10
50	Finger: Serial ASCII Word Generator	0 1 2 3 4 5 6 7 8 9 10
60	Nelson: "Chip" Off the Olde PDP-8/E	0 1 2 3 4 5 6 7 8 9 10
70	Frank: Analog/Digital Conversion Techniques	0 1 2 3 4 5 6 7 8 9 10
74	Jewell: Simplify Your Homemade Assembler	0 1 2 3 4 5 6 7 8 9 10

February BOMB Results

Winner of the $50 prize for the most popular article in the February 1976 issue of BYTE is W Douglas Maurer's "Processing Algebraic Expressions." Runners up in the voting were Robert Grappel's "My Dear Aunt Sally" and Don Lancaster's "Color TV Graphics."

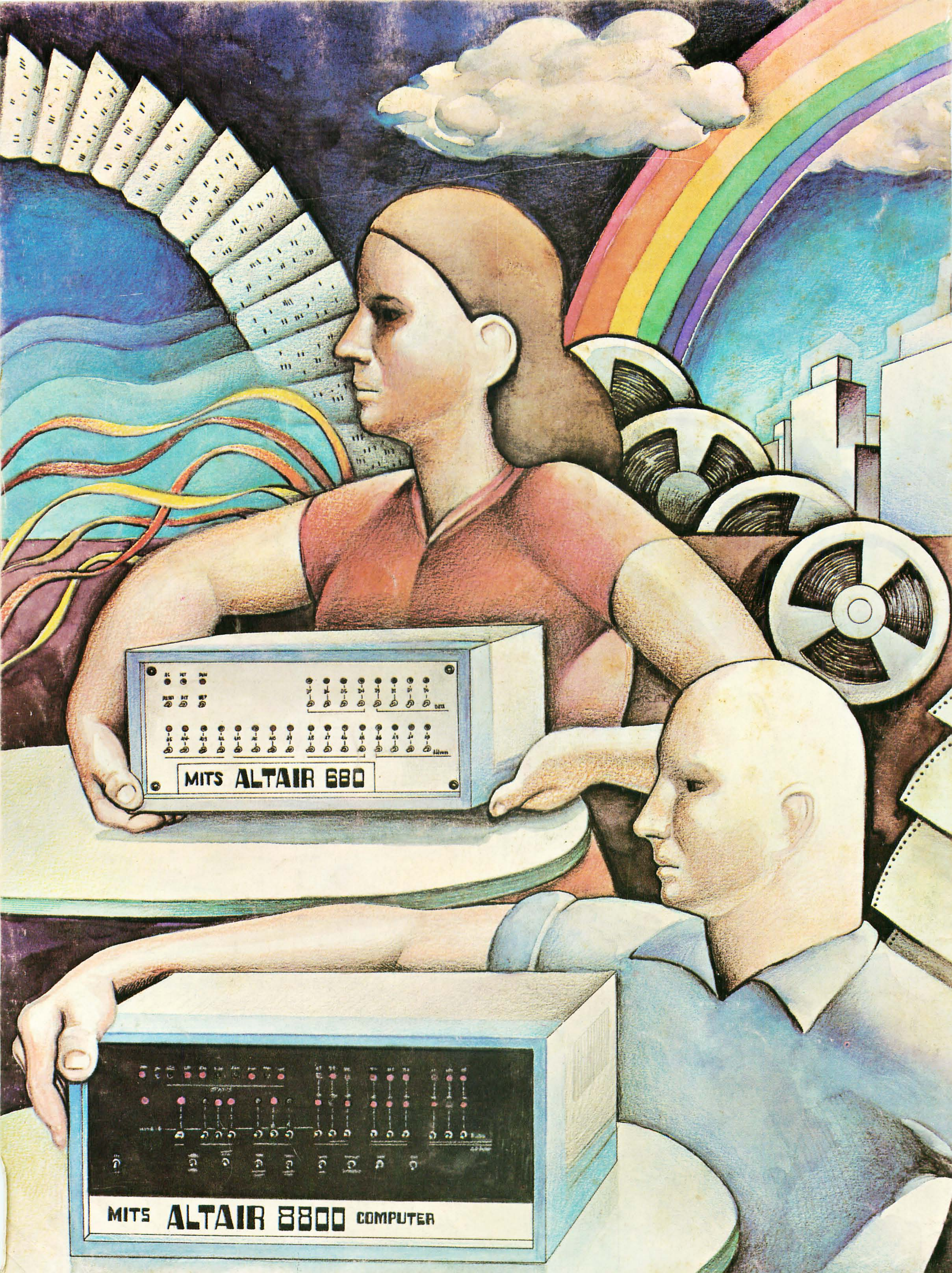
Feel free to photocopy this or any other page if you wish to keep your BYTE intact.

floppy disk in which compatibility with higher priced media format might be thrown out the window if necessary, and cost cutting is the primary goal. The approaches to cost cutting might include lowering data density to relax mechanical tolerances, use of slower access mechanisms, use of some manual operations in place of electronically controlled ones, etc. The engineering of such an inexpensive device I will leave to the inventors; the requirement of inexpensive random access mass storage is very real and will meet with a large market if the problem is solved.

Other approaches to the on line mass storage problem, such as bubble memories and CCD devices, may eventually be of some interest; but for the moment, cost is out of the consumer market's range, and non electrical off line copies are still needed when the power fails.

Personal Systems

The general trend in hardware and software which will lead to the most widespread availability of computing (and the largest future markets for companies who support this trend) is toward a smoothly packaged product which can be made to work with the minimum difficulty by any literate and thinking person. This is the black box approach to computing, one which the true computer hobbyist finds foreign, but which the person without special engineering talents or interests will purchase. When such personal systems become widely available to the general public, the application of computer technology to everyday problems will become widespread and large mass markets for applications programming products can begin to develop. I am talking here about a maturity of the small systems industry into a new equivalent of what happened at an earlier time with the automotive industry. I am also talking about a development of technological mass marketing which will have an impact similar to the automobile in its effects of opening up a multitude of new options for people through computing. There will always be the computer enthusiast market, just as there is now and has always been a "speed" market for automotive specialty products. The people who are heavily involved in hardware and software design and developing practical applications ideas will become the entrepeneurs and purveyors of products as the larger general market develops. Who will become the General Motors of the computing field? I can't predict that by any means. But I will predict that there will be such a concern. It will be an interesting show to watch as the next decade unfolds. ■


MITS ALTAIR 680
MITS ALTAIR 8800 COMPUTER